The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss

Simon Meister, Junhwa Hur, Stefan Roth
Department of Computer Science
TU Darmstadt, Germany

Abstract

In the era of end-to-end deep learning, many advances in computer vision are driven by large amounts of labeled data. In the optical flow setting, however, obtaining dense per-pixel ground truth for real scenes is difficult and thus such data is rare. Therefore, recent end-to-end convolutional networks for optical flow rely on synthetic datasets for supervision, but the domain mismatch between training and test scenarios continues to be a challenge. Inspired by classical energy-based optical flow methods, we design an unsupervised loss based on occlusion-aware bidirectional flow estimation and the robust census transform to circumvent the need for ground truth flow. On the KITTI benchmarks, our unsupervised approach outperforms previous unsupervised deep networks by a large margin, and is even more accurate than similar supervised methods trained on synthetic datasets alone. By optionally fine-tuning on the KITTI training data, our method achieves competitive optical flow accuracy on the KITTI 2012 and 2015 benchmarks, thus in addition enabling generic pre-training of supervised networks for datasets with limited amounts of ground truth.

Introduction

Estimating dense optical flow is one of the longstanding problems in computer vision, with a variety of applications. Though numerous approaches have been proposed over the past decades, *e.g*. (Black and Anandan 1991; Bruhn, Weickert, and Schnörr 2005; Brox and Malik 2011; Revaud et al. 2015), realistic benchmarks such as MPI Sintel (Butler et al. 2012) or KITTI (Geiger, Lenz, and Urtasun 2012; Menze and Geiger 2015) continue to challenge traditional energy minimization approaches. Motivated by the recent successes of end-to-end deep learning, convolutional neural networks (CNNs) have been suggested for addressing these challenges, and recently started to outperform many traditional flow methods in public benchmarks (Dosovitskiy et al. 2015; Gadot and Wolf 2016; Güney and Geiger 2016; Ilg et al. 2017).

Still, most CNN-based methods rely on the availability of a large amount of data with ground truth optical flow for supervised learning. Due to the difficulty of obtaining ground truth in real scenes, such networks are trained on synthetically generated images, for which dense ground truth is easier to obtain in large amounts (Dosovitskiy et al. 2015; Mayer et al. 2016). However, because of the intrinsic differences between synthetic and real imagery and the limited variability of synthetic datasets, the generalization to real scenes remains challenging. We posit that for realizing the potential of deep learning in optical flow, addressing this issue will be crucial.

In order to cope with the lack of labeled real-world training data, recent work (Ahmadi and Patras 2016; Yu, Harley, and Derpanis 2016; Ren et al. 2017) has proposed optical flow networks based on unsupervised learning. They sidestep the need for ground truth motion as only image sequences are required for training. Though unsupervised learning seems to be a promising direction for breaking the dependency on synthetic data, one may argue that these approaches fall short of the expectations so far, as they neither match nor surpass the accuracy of supervised methods despite the domain mismatch between training and test time.

In this paper, we introduce an end-to-end unsupervised approach that demonstrates the effectiveness of unsupervised learning for optical flow. Building on recent optical flow CNNs (Dosovitskiy et al. 2015; Ilg et al. 2017), we replace the supervision from synthetic data by an unsupervised photometric reconstruction loss similar to (Yu, Harley, and Derpanis 2016). We compute bidirectional optical flow (*i.e*., both in forward and backward direction, see Fig. 1) by performing a second pass with the two input images exchanged. Then, we design a loss function leveraging bidirectional flow to explicitly reason about occlusion (Hur and Roth 2017) and make use of the census transform to increase robustness on real images. Through a comprehensive ablation study, we validate the design choices behind our unsupervised loss.

On the challenging KITTI benchmarks (Geiger, Lenz, and Urtasun 2012; Menze and Geiger 2015), our unsupervised model outperforms previous unsupervised deep networks by a very large margin. Perhaps more surprisingly, it also surpasses architecturally similar supervised approaches trained exclusively on synthetic data. Similar observations hold, for example, on the Middlebury dataset (Baker et al. 2011). After unsupervised training on a large amount of realistic domain data, we can optionally make use of sparse ground truth (if available) to refine our estimates in areas that are inherently difficult to penalize in an unsupervised way, such as at motion boundaries. Our fine-tuned model achieves lead-

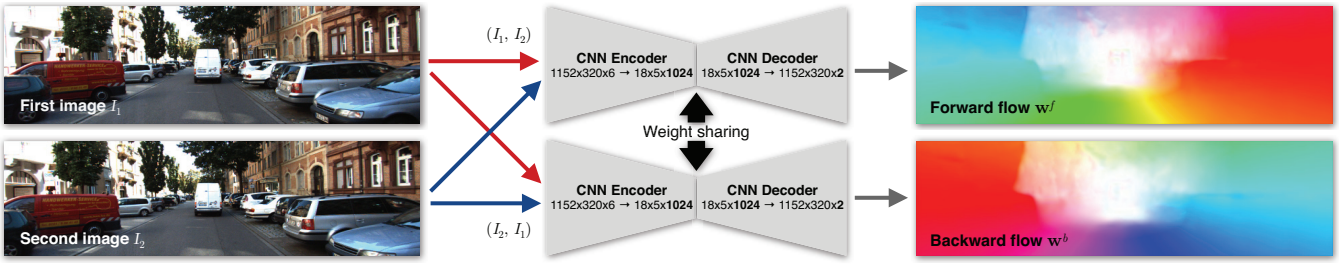

Figure 1: Bidirectional training of FlowNet (Dosovitskiy et al. 2015) with our comprehensive unsupervised loss.

ing accuracy on KITTI, thus demonstrating that our unsupervised learning approach also enables pre-training for supervised methods on domains with limited amounts of ground truth data. We thus present a step toward eliminating the need for careful engineering of synthetic datasets, as our networks can be trained on other domains with the expectation of achieving competitive accuracy.

Related Work

End-to-end supervised learning. End-to-end supervised learning of convolutional networks for optical flow was first introduced with FlowNet (Dosovitskiy et al. 2015). The network takes two consecutive input images and outputs a dense optical flow map using an encoder-decoder architecture. For supervised training, a large synthetic dataset was generated from static background images and renderings of animated 3D chairs. However, most likely due to the limited realism of the training dataset, FlowNet (Dosovitskiy et al. 2015) performs worse on the realistic KITTI 2012 benchmark (Geiger, Lenz, and Urtasun 2012) when compared to synthetic benchmarks like MPI Sintel (Butler et al. 2012).

A follow up work (Ilg et al. 2017) introduced the more accurate, but also slower and more complex FlowNet2 family of supervised networks. They improve upon the original architecture by stacking multiple FlowNet networks for iterative refinement. In addition to the original synthetic data, a more complex synthetic dataset (Mayer et al. 2016) featuring a larger variety of objects and motions is used for training. For the KITTI benchmark, FlowNet2 is fine-tuned on the sparse ground truth from the KITTI 2012 (Geiger, Lenz, and Urtasun 2012) and KITTI 2015 (Menze and Geiger 2015) training sets and achieves state-of-the-art accuracy.

A number of other deep architectures based on supervised learning have been proposed, leveraging various paradigms, including patch matching (Gadot and Wolf 2016), discrete optimization (Güney and Geiger 2016), and coarse-to-fine estimation (Ranjan and Black 2017). While we focus on the FlowNet architecture and its variants here, we note that our unsupervised loss can, in principle, be combined with other network architectures that directly predict the flow, *e.g.* (Ranjan and Black 2017).

End-to-end unsupervised learning. Recently, Yu, Harley, and Derpanis (2016) and Ren et al. (2017) suggested an unsupervised method based on FlowNet for sidestepping the limitations of synthetic datasets. As first proposed in earlier work with a different network architecture (Ahmadi and Patras 2016), they replace the original supervised loss by a proxy loss based on the classical brightness constancy and smoothness assumptions. The unsupervised network is trained on unlabeled image pairs from videos provided by the raw KITTI dataset (Geiger et al. 2013). On the KITTI 2012 benchmark (Geiger, Lenz, and Urtasun 2012), the unsupervised method fails to outperform the original FlowNet, probably due to the proxy loss being too simplistic. Subsequent work attempted to overcome these problems by combining the unsupervised proxy loss with proxy ground truth generated by a classical optical flow algorithm (Zhu et al. 2017). However, the gain on previous purely unsupervised approaches is minor, falling short of the original supervised FlowNetS. Other subsequent work (Zhu and Newsam 2017) attempted to improve unsupervised CNNs by replacing the underlying FlowNet architecture with a different network. Again, the method shows only little improvement over (Yu, Harley, and Derpanis 2016; Ren et al. 2017) and is still outperformed by the supervised FlowNetS.

As prior work does not come close to the accuracy of supervised methods, it remains unclear if unsupervised learning can overcome some of the limitations of supervised methods in realistic scenarios. In this paper, we therefore investigate possible ways for improving the accuracy of unsupervised approaches and aim to uncover whether they are a viable alternative or addition to supervised learning.

Unsupervised Learning of Optical Flow

We build on the previous FlowNetS-based UnsupFlownet (Yu, Harley, and Derpanis 2016) and extend it in three important ways. First, we design a symmetric, occlusion-aware loss based on bidirectional (*i.e.*, forward and backward) optical flow estimates. Second, we train FlowNetC with our comprehensive unsupervised loss to estimate bidirectional flow. Third, we make use of iterative refinement by stacking multiple FlowNet networks (Ilg et al. 2017). Optionally, we can also use a supervised loss for fine-tuning our networks on sparse ground truth data after unsupervised training.

Unsupervised loss

Let $I_1, I_2 : P \to \mathbb{R}^3$ be two temporally consecutive frames. Our goal is to estimate the optical flow $\mathbf{w}^f = (u^f, v^f)^T$ from I_1 to I_2. As our occlusion detection additionally requires the inverse (backward) optical flow $\mathbf{w}^b = (u^b, v^b)^T$,

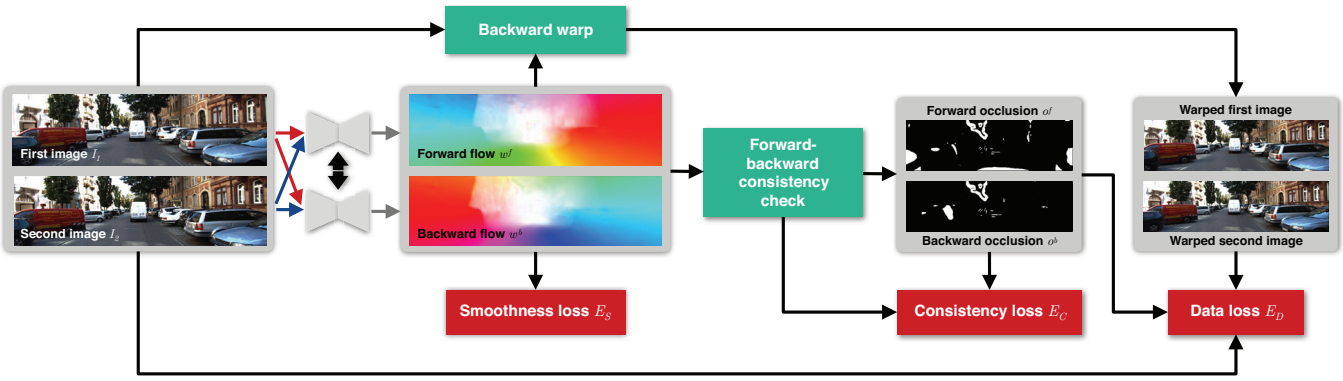

Figure 2: Schematic of our unsupervised loss. The data loss compares flow-warped images to the respective original images and penalizes their difference. We use forward-backward consistency based on warping the flow fields for estimating occlusion maps, which mask the differences in the data loss. Both flow fields are regularized assuming 2nd-order smoothness.

we jointly estimate bidirectional flow by making all of our loss terms symmetrical (*i.e.*, computing them for both flow directions). Bidirectional flow estimation taking into account occlusions has been proposed in the context of classical energy-based optical flow (Hur and Roth 2017). Here, we extend this idea from a superpixel-based setting to general flow fields and apply it as a loss function in unsupervised learning. Note that for brevity, some textual explanations discuss the forward direction only, however the same reasoning will apply to the backward direction as well.

Our unsupervised loss is based on the observation that a pixel in the first frame should be similar to the pixel in the second frame to which it is mapped by the flow (Fleet and Weiss 2006), which we encourage with our data loss. This observation does not hold for pixels that become occluded, however, as the corresponding pixels in the second frame are not visible. Thus, we mask occluded pixels from the data loss to avoid learning incorrect deformations that fill in the occluded pixels (Xiao et al. 2006). Our occlusion detection is based on the forward-backward consistency assumption (Sundaram, Brox, and Keutzer 2010). That is, for non-occluded pixels, the forward flow should be the inverse of the backward flow at the corresponding pixel in the second frame. We mark pixels as becoming occluded whenever the mismatch between these two flows is too large. Thus, for occlusion in the forward direction, we define the occlusion flag $o^f_{\mathbf{x}}$ to be 1 whenever the constraint

$$\left|\mathbf{w}^f(\mathbf{x}) + \mathbf{w}^b\big(\mathbf{x} + \mathbf{w}^f(\mathbf{x})\big)\right|^2 < \alpha_1 \left(\left|\mathbf{w}^f(\mathbf{x})\right|^2 + \left|\mathbf{w}^b\big(\mathbf{x} + \mathbf{w}^f(\mathbf{x})\big)\right|^2 \right) + \alpha_2 \quad (1)$$

is violated, and 0 otherwise. For the backward direction, we define $o^b_{\mathbf{x}}$ in the same way with $\mathbf{w}^f$ and $\mathbf{w}^b$ exchanged. We set $\alpha_1 = 0.01$, $\alpha_2 = 0.5$ in all of our experiments. As a baseline, we will also explore a loss variant with occlusion masking disabled. In that case, we simply let $o^f_{\mathbf{x}} = o^b_{\mathbf{x}} = 0$ for all $\mathbf{x} \in P$.

Our occlusion-aware data loss is now defined as

$$E_D(\mathbf{w}^f, \mathbf{w}^b, o^f, o^b) = \sum_{\mathbf{x}\in P} (1 - o^f_{\mathbf{x}}) \cdot \rho\Big(f_D\big(I_1(\mathbf{x}), I_2(\mathbf{x} + \mathbf{w}^f(\mathbf{x}))\big)\Big) + (1 - o^b_{\mathbf{x}}) \cdot \rho\Big(f_D\big(I_2(\mathbf{x}), I_1(\mathbf{x} + \mathbf{w}^b(\mathbf{x}))\big)\Big) + o^f_{\mathbf{x}}\lambda_p + o^b_{\mathbf{x}}\lambda_p, \quad (2)$$

where $f_D(I_1(\mathbf{x}), I_2(\mathbf{x}'))$ measures the photometric difference between two putatively corresponding pixels $\mathbf{x}$ and $\mathbf{x}'$, and $\rho(x) = (x^2 + \epsilon^2)^\gamma$ is the robust generalized Charbonnier penalty function (Sun, Roth, and Black 2014), with $\gamma = 0.45$. We add a constant penalty λ_p for all occluded pixels to avoid the trivial solution where all pixels become occluded, and penalize the photometric difference for all non-occluded pixels. In previous work (Yu, Harley, and Derpanis 2016), the brightness constancy constraint $f_D(I_1(\mathbf{x}), I_2(\mathbf{x}')) = I_1(\mathbf{x}) - I_2(\mathbf{x}')$ was used for measuring the photometric difference. As the brightness constancy is not invariant to illumination changes common in realistic situations (Vogel, Schindler, and Roth 2013), we instead use the ternary census transform (Zabih and Woodfill 1994; Stein 2004). The census transform can compensate for additive and multiplicative illumination changes as well as changes to gamma (Hafner, Demetz, and Weickert 2013), thus providing us with a more reliable constancy assumption for realistic imagery.

Unlike Yu, Harley, and Derpanis (2016), we use a second-order smoothness constraint (Trobin et al. 2008; Zhang et al. 2014) on the flow field to encourage collinearity of neighboring flows and thus achieve more effective regularization:

$$E_S(\mathbf{w}^f, \mathbf{w}^b) = \sum_{\mathbf{x}\in P} \sum_{(\mathbf{s},\mathbf{r})\in N(\mathbf{x})} \rho\Big(\mathbf{w}^f(\mathbf{s}) - 2\mathbf{w}^f(\mathbf{x}) + \mathbf{w}^f(\mathbf{r})\Big) + \rho\Big(\mathbf{w}^b(\mathbf{s}) - 2\mathbf{w}^b(\mathbf{x}) + \mathbf{w}^b(\mathbf{r})\Big), \quad (3)$$

where $N(\mathbf{x})$ consists of the horizontal, vertical, and both diagonal neighborhoods around $\mathbf{x}$ (4 in total). For vectorial

arguments, we assume that $\rho(\cdot)$ computes the average over the original generalized Charbonnier penalties of each component. Note that for occluded pixel positions, this term is the only one active besides the occlusion penalty.

For non-occluded pixels, we add a forward-backward consistency penalty

$$E_C(\mathbf{w}^f, \mathbf{w}^b, o^f, o^b) = \sum_{\mathbf{x}\in P} (1 - o^f_{\mathbf{x}}) \cdot \rho\Big(\mathbf{w}^f(\mathbf{x}) + \mathbf{w}^b\big(\mathbf{x} + \mathbf{w}^f(\mathbf{x})\big)\Big) + (1 - o^b_{\mathbf{x}}) \cdot \rho\Big(\mathbf{w}^b(\mathbf{x}) + \mathbf{w}^f\big(\mathbf{x} + \mathbf{w}^b(\mathbf{x})\big)\Big). \quad (4)$$

Then, our final loss is a weighted sum of the individual loss terms

$$E(\mathbf{w}^f, \mathbf{w}^b, o^f, o^b) = E_D(\mathbf{w}^f, \mathbf{w}^b, o^f, o^b) + \lambda_S E_S(\mathbf{w}^f, \mathbf{w}^b) + \lambda_C E_C(\mathbf{w}^f, \mathbf{w}^b, o^f, o^b). \quad (5)$$

The interplay of all losses is illustrated in Fig. 2.

Backward warping. To compute our losses in a subdifferentiable way for use with backpropagation, we employ bilinear sampling at flow-displaced positions (*i.e.*, backward warping). For example, to compare $I_1(\mathbf{x})$ and $I_2(\mathbf{x} + \mathbf{w}^f(\mathbf{x}))$, we backward-warp I_2 using $\mathbf{w}^f$ as described by Yu, Harley, and Derpanis (2016) and then compare the backward-warped second image to the first image. We use the bilinear sampling scheme of Jaderberg et al. (2015) to whom we refer for details.

Loss differentiation and implementation. We implement all losses as well as the warping scheme with primitive TensorFlow functions (Abadi and others 2015) and use automatic differentiation for backpropagation. Source code for training and evaluating our models is publicly available.

Network architecture and computation

UnFlow-C. Our basic CNN, termed *UnFlow-C*, is based on FlowNetC (Dosovitskiy et al. 2015), which processes two consecutive images in two separate input streams, explicitly correlates them, and compresses the result with a CNN encoder down to a sixth of the original resolution. In a decoder part (*refinement network*), the compressed representation is convolutionally upsampled four times as in FlowNetC, and dense flow is predicted after each upsampling. The last flow estimate is then bilinearly upsampled to the original resolution. To compute bidirectional optical flow, we first apply FlowNetC to the RGB images (I_1, I_2) to obtain the forward flow (u^f, v^f) and apply the same computation to (I_2, I_1) to obtain the backward flow (u^b, v^b). We share weights between both directions to train a universal network for optical flow in either direction (see Fig. 1).

Stacking. Inspired by FlowNet2 (Ilg et al. 2017), we iteratively refine the estimate of *UnFlow-C* by passing it into a separate FlowNetS with independent weights and term the two-network stack *UnFlow-CS*. Again, we perform one pass for each flow direction and share all weights between the two passes. In addition to the original images, we input the initial flow estimate, the backward-warped second image, and the brightness error between the warped image and the first image into the iterated network. In the same way, we concatenate an additional FlowNetS after UnFlow-CS to refine its estimate and term the three-network stack *UnFlow-CSS*.

i^{th} layer	Scale	Weight λ^i_l	Patch size
0	2^{-6}	1.1	3×3
1	2^{-5}	3.4	3×3
2	2^{-4}	3.9	5×5
3	2^{-3}	4.35	5×5
4	2^{-2}	12.7	7×7

Table 1: Loss settings to penalize flow fields predicted at different stages of the refinement network (for the final, highest resolution estimate, $i = 4$). The scale of the respective estimate is given as a fraction of the input image resolution and the loss weight decreases with the resolution of the estimate. We use a smaller patch size for the census transform as resolution decreases.

Computing the unsupervised loss. Similar to the supervised FlowNet, we compute losses for all intermediate predictions (*cf.* Fig. 3 in (Dosovitskiy et al. 2015) or Fig. 2 in (Yu, Harley, and Derpanis 2016)) from the refinement network to guide the learning process at multiple resolutions and then combine them by taking a weighted average. Our total loss is

$$E_{\text{unsup}} = \sum_i \lambda^i_l E_i, \quad (6)$$

where E_i is the loss from Eq. (5), evaluated at layer i. Table 1 details the settings we use for each individual layer.

Supervised loss for fine-tuning. For supervised fine-tuning on the KITTI training sets (only relevant for network variants with suffix *-ft*), we compute the network loss by comparing the bilinearly upsampled final flow estimate to the ground truth flow at all pixels for which ground truth is available:

$$E_{\text{sup}}(\mathbf{w}^f) = \sum_{\mathbf{x}\in P} v^f_{\mathbf{x}} \rho\big(\mathbf{w}^f(\mathbf{x}) - \mathbf{w}^f_{gt}(\mathbf{x})\big), \quad (7)$$

where $v^f_{\mathbf{x}} = 1$ if there is valid ground truth at pixel $\mathbf{x}$ and $v^f_{\mathbf{x}} = 0$ otherwise. As we want to avoid further increasing the sparsity of the ground truth, we do not downsample the ground truth flow and compute the loss for the final prediction only. Note that during fine-tuning, we only compute the first pass of the network for the forward flow direction, as there is ground truth for this direction only.

Discussion

As opposed to supervised approaches, which are often limited by the amount of available ground truth data, unsupervised techniques such as ours are limited by the loss function. That is, it should be noted that the amount of information that can be gained from the available video data is limited by how faithfully the problem is modeled by the loss. This is the key challenge that we are trying to address here.

Another limitation of our method is that we need to perform a parameter search for the weighting between the loss terms (*e.g.*, the influence of the smoothness loss) for best results, which increases the total computation time for training a model on a new domain for the first time. This limitation is shared by previous works employing an unsupervised proxy loss (Ahmadi and Patras 2016; Yu, Harley, and Derpanis 2016; Ren et al. 2017).

Compared to standard energy-based optical flow methods, our unsupervised networks avoid expensive optimization at test time. Moreover, stochastic minimization of the loss over a whole (training) dataset, as done here, can avoid some of the pitfalls of optimizing a complex energy on individual inputs, as done in classical approaches.

Datasets for Training

SYNTHIA. Ilg et al. (2017) showed that pre-training FlowNet on the synthetic FlyingChairs dataset before training on more complex and realistic datasets consistently improves the accuracy of their final networks. They conjecture that this pre-training can help the networks to learn general concepts of optical flow before learning to handle complex lighting and motion. Motivated by this, we pre-train our unsupervised models on the synthetic SYNTHIA dataset (Ros et al. 2016), for which no optical flow ground truth is available. The SYNTHIA dataset consists of multiple views recorded from a vehicle driving through a virtual environment. We use left images from the front, back, left, and right views of all winter and summer driving sequences, which amount to about 37K image pairs. Note that we can, of course, also pre-train on Flying Chairs (with comparable results after the full training procedure), but we specifically avoid this to show that we do not depend on this dataset.

KITTI. The KITTI dataset (Geiger et al. 2013) consists of real road scenes captured by a car-mounted stereo camera rig. A laser scanner provides accurate yet sparse optical flow ground truth for a small number of images, which make up the KITTI 2012 (Geiger, Lenz, and Urtasun 2012) and KITTI 2015 (Menze and Geiger 2015) flow benchmarks. In addition, a large dataset of raw 1392×512 image sequences is provided without ground truth. For unsupervised training, we use pairs of contiguous images from the city, residential, and road categories of the raw dataset. To avoid training on images on which we evaluate, we remove all images included in the KITTI 2012 and 2015 train and test sets from the raw dataset, including their temporal neighbors (within ± 10 frames). This final dataset consists of about 72K image pairs. For *optional* supervised fine-tuning, we use the 394 image pairs with ground truth from the KITTI 2012 and KITTI 2015 training sets. We evaluate our networks on the 195 testing and 194 training pairs of KITTI 2012 and the 200 testing and 200 training pairs of KITTI 2015.

Cityscapes. The Cityscapes dataset (Cordts et al. 2016) contains real driving sequences annotated for semantic segmentation and instance segmentation, without optical flow ground truth. In addition to our main experiments on KITTI, we also train a model on Cityscapes. We use all consecutive frames from the train/val/test sequences. As the KITTI benchmarks are at 10Hz, while Cityscapes is recorded at 17Hz, we also include all pairs where each second frame is skipped to learn larger displacements at 8.5Hz. The final dataset consists of about 230K image pairs.

Experiments

Training and evaluation details

Our networks are first trained on SYNTHIA and then on KITTI raw or Cityscapes in the proposed unsupervised way. We then *optionally* apply supervised fine-tuning to our best stacked networks trained on KITTI, for which we use ground truth from the KITTI 2012 and 2015 training sets. As optimizer, we use Adam (Kingma and Ba 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

Unsupervised SYNTHIA pre-training. We train for 300K iterations with a mini-batch size of 4 image pairs from the SYNTHIA data. We keep the initial learning rate of 1.0e–4 fixed for the first 100K iterations and then divide it by two after every 100K iterations. When training stacked networks, we pre-train each network while keeping the weights of any previous networks fixed.

Unsupervised KITTI training. We train for 500K iterations with a mini-batch size of 4 image pairs from the raw KITTI data. We keep the initial learning rate of 1.0e–5 fixed for the first 100K iterations and then divide it by two after every 100K iterations. For stacking, the same method as for SYNTHIA is applied.

Unsupervised Cityscapes training. The procedure is the same as for KITTI, expect that we only train a single network without stacking.

Supervised KITTI fine-tuning (*-ft*)**.** We fine-tune with a mini-batch size of 4 image pairs from the KITTI training sets and an initial learning rate of 0.5e–5, which we reduce to 0.25e–5 and 0.1e–5 after 45K and 65K iterations, respectively. For validation, we set aside 20% of the shuffled training pairs and fine-tune until the validation error increases, which generally occurs after about 70K iterations. Fine-tuning of a stack is done end-to-end.

Pre-processing and data augmentations during training. First, we shuffle the list of image pairs, and then randomly crop KITTI (both, raw and training) images to 1152×320, SYNTHIA images to 768×512, and Cityscapes images to 1024×512. Following Yu, Harley, and Derpanis (2016), we apply random additive Gaussian noise ($0 < \sigma \leq 0.04$), random additive brightness changes, random multiplicative color changes ($0.9 \leq$ multiplier ≤ 1.1), as well as contrast (from $[-0.3, 0.3]$) and gamma changes (from $[0.7, 1.5]$) to both frames independently. For unsupervised training only, we first apply the same random horizontal flipping and scaling ($0.9 \leq$ factor ≤ 1.1) to both frames, and finally a relative random scaling of the second frame ($0.9 \leq$ factor ≤ 1.1). The brightness changes are sampled from a Gaussian with $\sigma = 0.02$, while all other random values are uniformly sampled from the given ranges.

Figure 3: Visual comparison of our fully unsupervised UnFlow-C on the KITTI 2015 training set. We compare a baseline unsupervised loss akin to (Yu, Harley, and Derpanis 2016) (middle) to our best unsupervised loss (right). For each example, we show the ground truth and estimated flow (upper row), as well as input image overlay and flow error (lower row). The KITTI 2015 error map depicts correct estimates (≤ 3 px or $\leq 5\%$ error) in blue and wrong estimates in red tones.

Evaluation. Images from the KITTI flow benchmarks have a resolution of 1241×376, 1226×370, or 1242×375. As FlowNet needs both image dimensions to be divisible by 2^6, we bilinearly upsample these input images to 1280×384 for accuracy evaluation. The resulting 1280×384 flow estimates are then bilinearly downscaled to the original size for comparison with the ground truth flow maps. As distances of the original pixels change after resampling, we scale the components of the estimated flow according to the image downscaling factor in the respective directions. Note that just zero-padding the input images up to the next valid size introduces visible artifacts at the image boundaries and significantly increases the error. For other datasets, we use bilinear resampling analogously.

Unsupervised loss comparison

To compare the effect of individual unsupervised loss terms, we first pre-train UnFlow-C on SYNTHIA using the census loss, second-order smoothness, and disabling occlusion handling and forward-backward consistency. We then continue unsupervised training on KITTI for various loss settings.

Table 2 compares the accuracy of different unsupervised losses on the training portions of the KITTI benchmarks. First, note that when using the exact training protocol of Yu, Harley, and Derpanis (2016), our re-implementation of their simple brightness constancy baseline loss (with the small change of being formulated bidirectionally) already achieves higher accuracy than their original implementation (*cf.* Table 3), which we ascribe to implementation details.

We then observe the effect of each modification of the unsupervised loss terms over our baseline (Yu, Harley, and Derpanis 2016) re-implementation. Our census loss significantly improves upon the original brightness constancy loss ($\sim 35\%$ improvement), the second-order smoothness loss clearly outperforms the first-order one ($\sim 5\%$ improvement, $\sim 17\%$ outlier reduction), and occlusion masking combined with forward-backward consistency decreases the overall error further ($\sim 14\%$ improvement). The combination of all three innovations thus decreases the average endpoint error (AEE, average Euclidean distance between ground truth and estimated flow) by a significant margin compared to previous unsupervised training approaches (to less than $0.5\times$).

It is important to note here that the AEE is not known to the network during training; the only information used in training comes from our unsupervised loss in Eq. (5).

Figure 3 visually compares models trained with a baseline

Loss parameters						KITTI 2012			KITTI 2015		
f_D	occ. masking	E_S	λ_S	λ_p	λ_C	AEE (All)	AEE (NOC)	Fl-all	AEE (All)	AEE (NOC)	Fl-all
† brightness	×	1st	3.0	0.0	0.0	8.0	4.1	–	15.5	8.6	–
brightness	×	1st	3.0	0.0	0.0	7.20	3.42	31.93%	14.34	7.35	41.43%
brightness	×	2nd	3.0	0.0	0.0	7.11	3.03	28.10%	14.17	6.67	38.64%
census	×	1st	3.0	0.0	0.0	4.66	1.83	20.85%	10.24	4.64	31.33%
census	×	2nd	3.0	0.0	0.0	4.40	1.63	17.22%	9.49	**4.10**	29.20%
census	✓	2nd	3.0	12.4	0.0	4.32	1.62	17.14%	9.17	4.40	29.21%
census	✓	2nd	3.0	12.4	0.2	**3.78**	**1.58**	**16.44%**	**8.80**	4.29	**28.95%**

Table 2: Comparison of different loss terms on the training sets of the KITTI benchmarks. AEE: Average Endpoint Error; Fl-all: Ratio of pixels where flow estimate is wrong by both ≥ 3 pixels and $\geq 5\%$. We report the AEE over all pixels (All) and over non-occluded pixels only (NOC). The best result for each metric is printed in bold. †For the first row only, we train following the schedule of Yu, Harley, and Derpanis (2016), with pre-training done on FlyingChairs instead of SYNTHIA.

Method	KITTI 2012				KITTI 2015			Middlebury		Sintel Final	
	AEE (All)		AEE (NOC)		AEE (All)	Fl-all		AEE		AEE	
	train	*test*	*train*	*test*	*train*	*train*	*test*	*train*	*test*	*train*	*test*
DDF (Güney and Geiger 2016)	–	3.4	–	1.4	–	–	21.17%	–	–	–	5.73
PatchBatch (Gadot and Wolf 2016)	–	3.3	–	1.3	–	–	21.07%	–	–	–	5.36
FlowFieldCNN (Bailer, Varanasi, and Stricker 2017)	–	3.0	–	1.2	–	–	18.68%	–	–	–	–
ImpPB+SPCI (Schuster, Wolf, and Gadot 2017)	–	2.9	–	1.1	–	–	17.78%	–	–	–	–
SDF (Bai et al. 2016)	–	2.3	–	1.0	–	–	11.01%	–	–	–	–
FlowNetS+ft (Dosovitskiy et al. 2015)	7.5	9.1	5.3	5.0	–	–	–	0.98	–	(4.44)	7.76
UnsupFlownet (Yu, Harley, and Derpanis 2016)	11.3	9.9	4.3	4.6	–	–	–	–	–	–	–
DSTFlow(KITTI) (Ren et al. 2017)	10.43	12.4	3.29	4.0	16.79	36 %	39 %	–	–	7.95	11.80
FlowNet2-C (Ilg et al. 2017)	–	–	–	–	11.36	–	–	–	–	–	–
FlowNet2-CSS (Ilg et al. 2017)	3.55	–	–	–	8.94	29.77%†	–	0.44	–	3.23	–
FlowNet2-ft-kitti (Ilg et al. 2017)	(1.28)	1.8	–	1.0	(2.30)	(8.61%)†	10.41%	0.56	–	4.66	–
UnFlow-C-Cityscapes (ours)	5.08	–	2.12	–	10.78	33.89%	–	0.85	–	8.23	–
UnFlow-C (ours)	3.78	–	1.58	–	8.80	28.94%	–	0.88	–	8.64	–
UnFlow-CS (ours)	3.30	–	1.26	–	8.14	23.54%	–	0.65	–	7.92	–
UnFlow-CSS (ours)	3.29	–	1.26	–	8.10	23.27%	–	0.65	–	7.91	10.22
UnFlow-CS-ft (ours)	(1.32)	1.9	(0.75)	0.9	(2.25)	(9.24%)	12.55%	0.64	–	11.99	–
UnFlow-CSS-ft (ours)	(1.14)	1.7	(0.66)	0.9	(1.86)	(7.40%)	11.11%	0.64	0.76	13.65	–

Table 3: Accuracy comparison on KITTI, Middlebury, and Sintel optical flow benchmarks. AEE: Average Endpoint Error; Fl-all: Ratio of pixels where flow estimate is wrong by both ≥ 3 pixels and $\geq 5\%$. The numbers in parentheses are the results of the networks on data they were trained on, and hence are not directly comparable to other results. †*train* numbers are quoted from Ilg et al. (2017), published before the recent, small update of the KITTI 2015 ground truth. For example, with this update the *test* Fl-all for FlowNet2-ft-kitti changed from 11.48% to 10.41%.

loss akin to (Yu, Harley, and Derpanis 2016) (1st and 2nd row of Table 2) and our best unsupervised loss (last row).

Results on benchmarks

We use the best loss setup from the previous ablation study for unsupervised training of our final networks and refer to our single FlowNetC, two-network stack, and three-network stack trained on SYNTHIA and KITTI images as UnFlow-C, UnFlow-CS, and UnFlow-CSS, respectively. Finally, we fine-tune UnFlow-CS and UnFlow-CSS with the supervised KITTI schedule and refer to these models as UnFlow-CS-ft and UnFlow-CSS-ft. In addition, we train a FlowNetC on SYNTHIA and Cityscapes and refer to this as UnFlow-C-Cityscapes. Table 3 compares the accuracy of our method and other optical flow methods on KITTI 2012, KITTI 2015, Middlebury, and MPI Sintel. Figure 4 visually compares FlowNet-based methods on KITTI 2012.

KITTI. UnsupFlownet (Yu, Harley, and Derpanis 2016) and DSTFlow (Ren et al. 2017) gave some improvements over the supervised FlowNetS (Dosovitskiy et al. 2015) in non-occluded regions at the cost of a significant increase in total endpoint error on KITTI 2012. On KITTI 2012, our purely unsupervised UnFlow-C outperforms the *supervised* FlowNetC and FlowNetS in all metrics. This highlights the benefit of being able to train on the relevant domain, which is possible with our approach even when no ground truth flow is available. When compared to UnsupFlownet and DSTFlow, UnFlow-C more than halves both error metrics on KITTI 2012 and strongly improves accuracy on KITTI 2015. Even our unsupervised UnFlow-C-Cityscapes, which was not trained on KITTI images, significantly outperforms UnsupFlownet and DSTFlow. This shows how our bidirectional loss based on a robust data loss significantly improves over previous unsupervised approaches. Finally, we observe

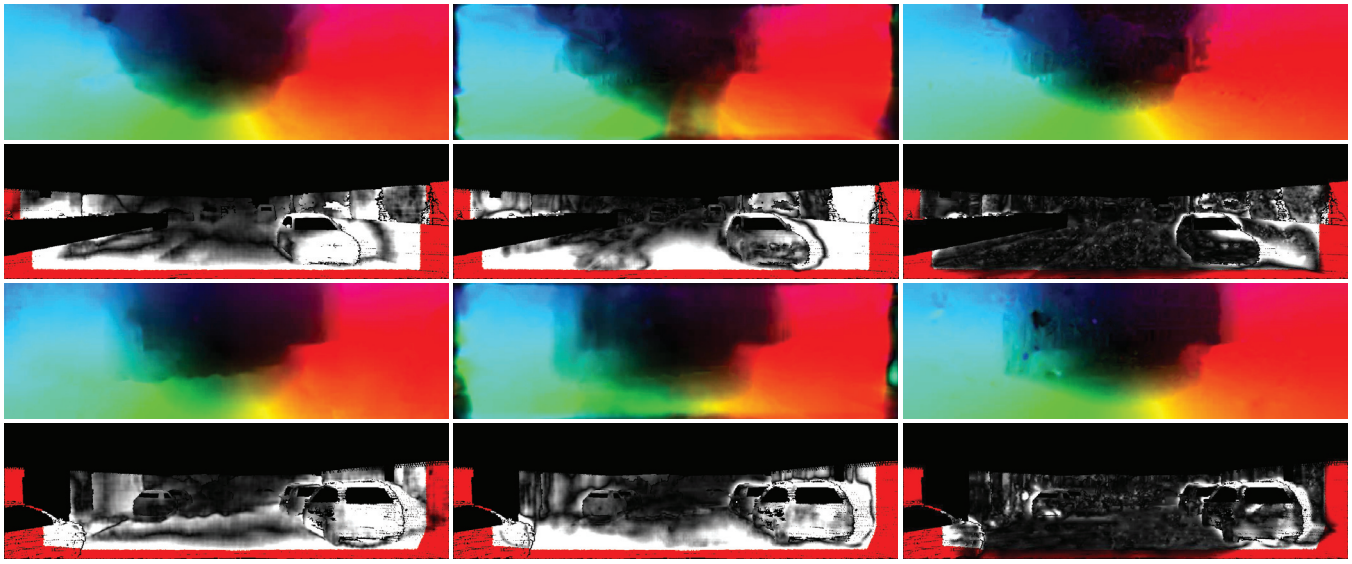

Figure 4: Visual comparison on the KITTI 2012 test set: (left to right) FlowNetS (Dosovitskiy et al. 2015), UnsupFlownet (Yu, Harley, and Derpanis 2016), and our unsupervised UnFlow-CSS. For each example, we show the flow (upper row) and flow error (lower row) maps. The KITTI 2012 error map scales linearly between 0 (black) and ≥ 5 pixels error (white).

that our fine-tuned network performs similar to the more complex FlowNet2-ft-kitti (Ilg et al. 2017) without the need for a separate small-displacement network and custom training schedules. Note that we do not make use of the extended synthetic dataset of (Mayer et al. 2016), but rely on the datasets only as described above. Moreover, our purely unsupervised networks clearly outperform the supervised FlowNet2 counterparts before fine-tuning. This again highlights the benefit of unsupervised learning of optical flow for coping with real-world domains.

Middlebury. On Middlebury, UnFlow-C and UnFlow-C-Cityscapes outperform the supervised FlowNetS, and our stacked and fine-tuned variants perform between FlowNetS and the complex FlowNet2 models. This demonstrates that our networks generalize to realistic domains outside the driving setting they were trained on.

Sintel. On Sintel, our unsupervised networks cannot compete with FlowNetS+ft and FlowNet2, which in contrast are trained on various datasets in a supervised manner. However, our method outperforms DSTFlow (Ren et al. 2017), which is also trained on a similar dataset in an unsupervised manner. This shows that our method not only strongly outperforms previous unsupervised deep networks for in-domain data, but also yields benefits for data from a domain (here, synthetic) it has not been trained on.

Conclusion

We presented an end-to-end unsupervised learning approach to enable effective training of FlowNet networks on large datasets for which no optical flow ground truth is available. To that end, we leveraged components from well-proven energy-based flow approaches, such as a data loss based on the census transform, higher-order smoothness, as well as occlusion reasoning enabled by bidirectional flow estimation. Our experiments show that using an accurate unsupervised loss, as proposed, is key to exploiting unannotated datasets for optical flow, more than halving the error on the challenging KITTI benchmark compared to previous unsupervised deep learning approaches. Consequently, we make CNNs for optical flow applicable to a larger range of domains. Our results, moreover, demonstrate that using a large, real-world dataset together with our unsupervised loss can even outperform supervised training on challenging realistic benchmarks when only handcrafted synthetic datasets are available for supervision. Finally, we showed that our unsupervised loss provides a solid foundation for pre-training when only limited amounts of real-world ground truth data are available. Going forward, our results suggest that further research on more accurate losses for unsupervised deep learning may be a promising direction for advancing the state-of-the-art in optical flow estimation.

Acknowledgements

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC Grant Agreement No. 307942.

References

Abadi, M., et al. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Ahmadi, A., and Patras, I. 2016. Unsupervised convolutional neural networks for motion estimation. In *ICIP*.

Bai, M.; Luo, W.; Kundu, K.; and Urtasun, R. 2016. Ex-

ploiting semantic information and deep matching for optical flow. In *ECCV*, volume 4, 154–170.

Bailer, C.; Varanasi, K.; and Stricker, D. 2017. CNN-based patch matching for optical flow with thresholded hinge loss. In *CVPR*.

Baker, S.; Scharstein, D.; Lewis, J. P.; Roth, S.; Black, M. J.; and Szeliski, R. 2011. A database and evaluation methodology for optical flow. *Int. J. Comput. Vision* 92(1):1–31.

Black, M. J., and Anandan, P. 1991. Robust dynamic motion estimation over time. In *CVPR*, 296–302.

Brox, T., and Malik, J. 2011. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE T. Pattern Anal. Mach. Intell.* 33(3):500–513.

Bruhn, A.; Weickert, J.; and Schnörr, C. 2005. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *Int. J. Comput. Vision* 61(3):211–231.

Butler, D. J.; Wulff, J.; Stanley, G. B.; and Black, M. J. 2012. A naturalistic open source movie for optical flow evaluation. In *ECCV*, volume 4, 611–625.

Cordts, M.; Omran, M.; Ramos, S.; Scharwächter, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 3213–3223.

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; van der Smagt, P.; Cremers, D.; and Brox, T. 2015. FlowNet: Learning optical flow with convolutional networks. In *ICCV*, 2758–2766.

Fleet, D. J., and Weiss, Y. 2006. Optical flow estimation. In Paragios, N.; Chen, Y.; and Faugeras, O., eds., *Handbook of Mathematical Models in Computer Vision*. Springer.

Gadot, D., and Wolf, L. 2016. PatchBatch: A batch augmented loss for optical flow. In *CVPR*.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* 32(11):1231–1237.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*.

Güney, F., and Geiger, A. 2016. Deep discrete flow. In *ACCV*.

Hafner, D.; Demetz, O.; and Weickert, J. 2013. Why is the census transform good for robust optic flow computation? In *International Conference on Scale Space and Variational Methods in Computer Vision*, 210–221.

Hur, J., and Roth, S. 2017. MirrorFlow: Exploiting symmetries in joint optical flow and occlusion estimation. In *ICCV*.

Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; and Brox, T. 2017. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; and Kavukcuoglu, K. 2015. Spatial transformer networks. In *NIPS*, 2017–2025.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; and Brox, T. 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*.

Menze, M., and Geiger, A. 2015. Object scene flow for autonomous vehicles. In *CVPR*.

Ranjan, A., and Black, M. J. 2017. Optical flow estimation using a spatial pyramid network. In *CVPR*.

Ren, Z.; Yan, J.; Ni, B.; Liu, B.; Yang, X.; and Zha, H. 2017. Unsupervised deep learning for optical flow estimation. In *AAAI Conference on Artificial Intelligence*.

Revaud, J.; Weinzaepfel, P.; Harchaoui, Z.; and Schmid, C. 2015. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In *ICCV*, 1164–1172.

Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; and Lopez, A. 2016. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*.

Schuster, T.; Wolf, L.; and Gadot, D. 2017. Optical flow requires multiple strategies (but only one network). In *CVPR*.

Stein, F. 2004. Efficient computation of optical flow using the census transform. In *DAGM*, 79–86.

Sun, D.; Roth, S.; and Black, M. J. 2014. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *Int. J. Comput. Vision* 106(2):115–137.

Sundaram, N.; Brox, T.; and Keutzer, K. 2010. Dense point trajectories by GPU-accelerated large displacement optical flow. In *ECCV*, volume 2, 438–451.

Trobin, W.; Pock, T.; Cremers, D.; and Bischof, H. 2008. An unbiased second-order prior for high-accuracy motion estimation,. In *DAGM*, 396–405.

Vogel, C.; Schindler, K.; and Roth, S. 2013. An evaluation of data costs for optical flow. In *GCPR*, 343–353.

Xiao, J.; Cheng, H.; Sawhney, H.; Rao, C.; and Isnardi, M. 2006. Bilateral filtering-based optical flow estimation with occlusion detection. In *ECCV*, 211–224.

Yu, J. J.; Harley, A. W.; and Derpanis, K. G. 2016. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *ECCV 2016 Workshops*, 3–10.

Zabih, R., and Woodfill, J. 1994. Non-parametric local transforms for computing visual correspondence. In *ECCV*, 151–158.

Zhang, C.; Li, Z.; Cai, R.; Chao, H.; and Rui, Y. 2014. As-rigid-as-possible stereo under second order smoothness priors. In *ECCV*, volume 2, 112–126.

Zhu, Y., and Newsam, S. D. 2017. DenseNet for dense flow. In *ICIP*.

Zhu, Y.; Lan, Z.; Newsam, S.; and Hauptmann, A. G. 2017. Guided optical flow learning. In *CVPR 2017 Workshops*.