Conditional Prior Networks for Optical Flow

Yanchao Yang(✉) and Stefano Soatto

UCLA Vision Lab, University of California, Los Angeles, CA 90095, USA
{yanchao.yang,soatto}@cs.ucla.edu

Abstract. Classical computation of optical flow involves generic priors (regularizers) that capture rudimentary statistics of images, but not long-range correlations or semantics. On the other hand, fully supervised methods learn the regularity in the annotated data, without explicit regularization and with the risk of overfitting. We seek to learn richer priors on the set of possible flows that are statistically compatible with *an* image. Once the prior is learned in a supervised fashion, one can easily learn the full map to infer optical flow directly from *two or more* images, without any need for (additional) supervision. We introduce a novel architecture, called Conditional Prior Network (CPN), and show how to train it to yield a conditional prior. When used in conjunction with a simple optical flow architecture, the CPN beats all variational methods and all unsupervised learning-based ones using the same data term. It performs comparably to fully supervised ones, that however are fine-tuned to a particular dataset. Our method, on the other hand, performs well even when transferred between datasets. Code is available at: https://github.com/YanchaoYang/Conditional-Prior-Networks.

1 Introduction

Consider Fig. 1: A given image (left) could give rise to many different optical flows (OF) [18] depending on what another image of the same scene looks like: It could show a car moving to the right (top), or the same apparently moving to the left due to camera motion to the right (middle), or it could be an artificial motion because the scene was a picture portraying the car, rather than the actual physical scene. A single image biases, but does not constrain, the set of possible flows the underlying scene can generate. We wish to leverage the information an image contains about possible compatible flows to learn better priors than those implied by generic regularizers. Note that all three flows in Fig. 1 are equally valid under a generic prior (piecewise smoothness), but not under a natural prior (cars moving in the scene).

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01267-0_17) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11219, pp. 282–298, 2018.
https://doi.org/10.1007/978-3-030-01267-0_17

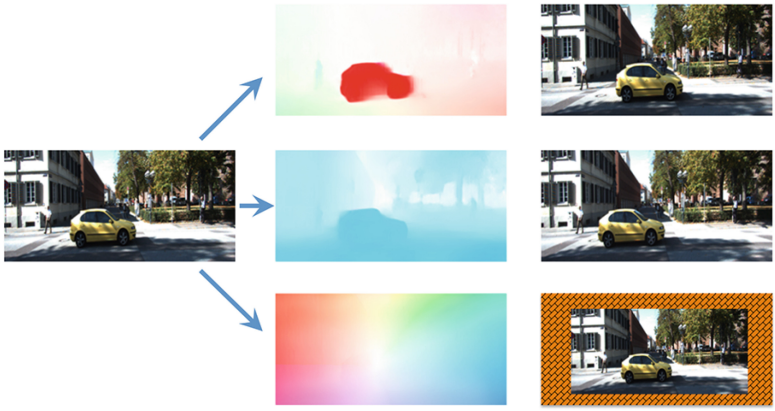

Fig. 1. A single image biases, but does not constrain, the set of optical flows that can be generated from it, depending on whether the camera was static but objects were moving (top), or the camera was moving (center), or the scene was flat (bottom) and moving on a plane in an un-natural scenario. Flow fields here are generated by our CPNFlow

A regularizer is a criterion that, when added to a data fitting term, constrains the solution of an inverse problem. These two criteria (data term and regularizer) are usually formalized as an energy function, which is minimized to, ideally, find a unique global optimum.[1]

1.1 Our Approach in Context

In classical (variational) OF, the regularizer captures very rudimentary low-order statistics [4,5,9,29,37], for instance the high kurtosis of the gradient distribution. This does not help with the scenario in Fig. 1. There has been a recent surge of (supervised) learning-based approaches to OF [15,19,32], that do not have explicit regularization nor do they use geometric reprojection error as a criterion for data fit. Instead, a map is learned from pairs of images to flows, where regularization is implicit in the function class [13],[2] in the training procedure [11] (e.g. noise of stochastic gradient descent – SGD), and in the datasets used for training (e.g. Sintel [10], Flying Chair [15]).

Our method does not attempt to learn geometric optics anew, even though black-box approaches are the top performers in several benchmarks. Instead, we seek to learn richer priors on the set of possible flows that are statistically compatible with an image (Fig. 1).

[1] We use the terms regularizer, prior, model, or assumption, interchangeably and broadly to include any restriction on the solution space, or bias on the solution, imposed without full knowledge of the data. In OF, the full data is (at least) two images.

[2] In theory, deep neural networks are universal approximants, but there is a considerable amount of engineering in the architectures to capture suitable inductive biases.

Unsupervised learning-based approaches use the same or similar loss functions as variational methods [2,20,27,33], including priors, but restrict the function class to a parametric model, for instance convolutional neural networks (CNNs) trained with SGD, thus adding implicit regularization [11]. Again, the priors only encode first-order statistics, which fail to capture the phenomena in Fig. 1.

We advocate learning a conditional prior, or regularizer, from data, but do so once and forall, and then use it in conjunction with any data fitting term, with any model and optimization one wishes.

What we learn is a prior in the sense that it imposes a bias on the possible solutions, but it does not alone constraint them, which happens only in conjunction with a data term. Once the prior is learned, in a supervised fashion, one can also learn the full map to infer optical flow directly from data, without any need for (additional) supervision. In this sense, our method is *"semi-unsupervised"*: Once *we* learn the prior, *anyone* can train an optical flow architecture entirely unsupervised. The key idea here is to learn a prior for the set of optical flows that are statistically compatible with *a single image*. Once done, we train a relatively simple network *in an unsupervised fashion* to map *pairs of images* to optical flows, where the loss function used for training includes explicit regularization in the form of the conditional prior, added to the reprojection error.

Despite a relatively simple architecture and low computational complexity, our method beats all variational ones and all unsupervised learning-based ones. It is on par or slightly below a few fully supervised ones, that however are fine-tuned to a particular dataset, and are extremely onerous to train. More importantly, available fully supervised methods perform best *on the dataset on which they are trained.* Our method, on the other hand, performs well even when the prior is trained on one dataset and used on a different one. For instance, a fully-supervised method trained on Flying Chair beats our method on Flying Chair, but underperforms it on KITTI and vice-versa (Table 1). Ours is consistently among the top in all datasets. More importantly, our method is complementary, and can be used in conjunction with more sophisticated networks and data terms.

1.2 Formalization

Let $I_1, I_2 \in \mathbb{R}_+^{H\times W\times 3}$ be two consecutive images and $f : \mathbb{R}^2 \to \mathbb{R}^2$ the flow, implicitly defined in the co-visible region by $I_1 = I_2 \circ f + n$ where $n \sim P_n$ is some distribution. The posterior $P(f|I_1, I_2) \propto P_n(I_1 - I_2 \circ f)$ can be decomposed as

$$\begin{aligned}\log P(f|I_1, I_2) &= \log P(I_2|I_1, f) + \log P(f|I_1) - \log P(I_2|I_1) \\ &\approx \log P(I_2|I_1, f) + \log P(f|I_1) \quad (1)\end{aligned}$$

We call the first term (data) **prediction error**, and the second **conditional prior**. It is a prior in the sense that, given I_1 alone, many flows can have high likelihood for a suitable I_2. However, it is informed by I_1 in the sense of capturing image-dependent regularities such as flow discontinuities often occurring at *object*

boundaries, which may or may not correspond to generic image discontinuities. A special case of this model assumes a Gaussian likelihood (ℓ^2 prediction error) and an ad-hoc prior of the form

$$E(f, I_1, I_2) = \int (I_1(x) - I_2(x + f(x)))^2 dx + \int \alpha(x, I_1) \|\nabla f(x)\|^2 dx \quad (2)$$

where α is a scalar function that incorporates our belief in an irradiance boundary of I_1 corresponding to an object boundary.[3] This type of conditional prior has several limitations: First, in the absence of *semantic context*, it is not possible to differentiate occluding boundaries (where f can be discontinuous) from material boundaries (irradiance discontinuities), or illumination boundaries (cast shadows) where f is smooth. Second, the image I_1 only informs the flow *locally*, through its gradient, and does not capture global regularities. Figure 2 shows that flow fails to propagate into homogeneous region. This can be mitigated by using a fully connected CRF [36] but at a heavy computational cost.

Fig. 2. First row: two images I_1, I_2 from the Flying Chair dataset; Second row: warped image $I_2 \circ \hat{f}$ (left) using the flow (right) estimated by minimizing Eq. (2); Third row: residual $n = \|I_1 - I_2 \circ f\|$ (left) compared to the edge strength of I_1 (right). Note the flow estimated at the right side of the chair fails to propagate into the homogeneous region where the image gradient is close to zero.

Our goal can be formalized as *learning the conditional prior* $P(f|I_1)$ in a manner that exploits the semantic context of the scene[4] and captures the global

[3] When α is constant, we get an even more special case, the original Horn & Schunk model where the prior is also Gaussian and unconditional (independent of I_1).

[4] The word "semantic" is often used to refer to *identities* and *relations* among discrete entities (objects). What matters in our case is the *geometric and topological relations*

statistics of I_1. We will do so by leveraging the power of deep convolutional neural networks trained end-to-end, to enable which we need to design differentiable models, which we do next.

2 Method

To learn a conditional prior we need to specify the inference criterion (loss function), which we do in Sect. 2.2 and the class of functions (architecture), with respect to which the loss is minimized end-to-end. We introduce our choice of architecture next, and the optimization in Sect. 2.3.

2.1 Conditional Prior Network (CPN)

We construct the conditional prior from a modified autoencoder trained to reconstruct a flow f that is compatible with the given (single) image I. We call this a Conditional Prior Network (CPN) shown in Fig. 3.

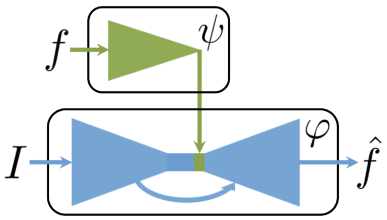

Fig. 3. Conditional Prior Network (CPN) architecture for learning $P(f|I)$: ψ is an encoder of the flow f, and φ is a decoder that has full access to the image I.

In a CPN, ψ encodes only the flow f, then φ takes the image I and the output of ψ to generate a reconstruction of f, $\hat{f} = \varphi(I, \psi(f))$. Both ψ and φ are realized by pure convolutional layers with subsampling (striding) by two to create a bottleneck. Note that φ is a U-shape net [15] with skip connections, at whose center a concatenation with $\psi(f)$ is applied. In the appendix, we articulate the reasons for our choice of architecture, and argue that it is better than an ordinary autoencoder that encodes both f and I in one branch. This is connected to the choice of loss function and how it is trained, which we discuss next.

2.2 Loss Function

We are given a dataset D sampled from the joint distribution $D = \{(f_j, I_j)\}_{j=1}^n \sim P(f, I)$, with n samples. We propose approximating $P(f|I)$ with a CPN as follows

$$Q_{w_\varphi, w_\psi}(f|I) = \exp\left(-\|\varphi(I, \psi(f)) - f\|^2\right) \propto P(f|I) \tag{3}$$

that may result in occluding boundaries on the image plane. The name of an object does not matter to that end, so we ignore identities and do not require object labels.

where w_φ, w_ψ are the parameters of φ and ψ respectively. Given I, for every flow f, the above returns a positive value whose log, after training, is equal to the negative squared autoencoding loss. To determine the parameters that yield an approximation of $P(f|I)$, we should solve the following optimization problem

$$w_\varphi^*, w_\psi^* = \arg\min_{w_\varphi, w_\psi} \mathbb{E}_{I\sim P(I)} \mathbb{KL}(P(f|I) \| Q_{w_\varphi, w_\psi}(f|I)) \tag{4}$$

where the expectation is with respect to all possible images I, and $\mathbb{KL}$ is the Kullback-Leibler divergence between $P(f|I)$ and the CPN $Q_{w_\varphi, w_\psi}(f|I)$. In the appendix we show that the above is equivalent to:

$$\begin{aligned} w_\varphi^*, w_\psi^* &= \arg\max_{w_\varphi, w_\psi} \int_I \int_f P(f, I) \log[Q_{w_\varphi, w_\psi}(f|I)] df dI \\ &= \arg\min_{w_\varphi, w_\psi} \int_I \int_f P(f, I) \| \varphi_{w_\varphi}(I, \psi_{w_\psi}(f)) - f \|^2 df dI \end{aligned} \tag{5}$$

which is equivalent to minimizing the empirical autoencoding loss since the ground truth flow is quantized, $\sum_{j=1}^n \| \hat{f}_j - f_j \|^2$. If the encoder had no bottleneck (sufficient information capacity), it could overfit by returning $\hat{f} = \varphi_{w_\varphi}(I, \psi_{w_\psi}(f)) = f$, rendering the conditional prior $Q_{w_\varphi, w_\psi}(f|I)$ uninformative (constant). Consistent with recent developments in the theory of Deep Learning [1], sketched in the appendix, we introduce an information regularizer (bottleneck) on the encoder ψ leading to the **CPN training loss**

$$w_\varphi^*, w_\psi^* = \arg\min_{w_\varphi, w_\psi} \mathbb{E}_{I\sim P(I)} \mathbb{KL}(P(f|I) \| Q_{w_\varphi, w_\psi}(f|I)) + \beta \boldsymbol{I}(f, \psi_{w_\psi}(f)) \tag{6}$$

where $\beta > 0$ modulates complexity (information capacity) and fidelity (data fit), and $\boldsymbol{I}(f, \psi_{w_\psi}(f))$ is the mutual information between the flow f and its representation (code) $\psi_{w_\psi}(f)$. When β is large, the encoder is lossy, thus preventing $Q_{w_\varphi, w_\psi}(f|I)$ from being uninformative.[5]

2.3 Training a CPN

While the first term in Eq. (6) can simply be the empirical autoencoding loss, the second term can be realized in many ways, *e.g.*, an ℓ^2 or ℓ^1 penalty on the parameters w_ψ. Here we directly increase the bottleneck β by decreasing the coding length ℓ_ψ of ψ. Hence the training procedure of the proposed CPN can be summarized as follows:

1. Initialize the coding length of the encoder ℓ_ψ with a large number ($\beta = 0$).
2. Train the encoder-decoder ψ, φ jointly by minimizing $e = \frac{1}{n} \sum_{j=1}^n \| \hat{f}_j - f_j \|^2$ until convergence. The error at convergence is denoted as e^*.
3. If $e^* > \lambda$, training done.[6]
   Otherwise, decrease ℓ_ψ, (increase β), and goto step 2.

[5] The decoder φ imposes no architectural bottleneck due to skip connections.
[6] In our experiments, $\lambda = 0.5$.

It would be time consuming to train for every single coding length ℓ_ψ. We only iteratively train for the integer powers, $2^k, k \leq 10$.

Inference: suppose the optimal parameters obtained from the training procedure are w^*_ψ, w^*_φ, then for any given pair (f, I), we can use $Q_{w^*_\varphi, w^*_\psi}(f|I)$ as the conditional prior up to a constant. In the next section we add a data discrepancy term to the (log) prior to obtain an energy functional for learning direct mapping from images to optical flows.

2.4 Semi-unsupervised Learning Optical Flow

Unlike a generative model such as a variational autoencoder [22], where sampling is required in order to evaluate the probability of a given observation, here (f, I) is directly mapped to a scalar using Eq. (3), thus differentiable w.r.t f, and suitable for training a new network to predict optical flow given images I_1, I_2, by minimizing the following compound loss:

$$\begin{aligned} E(f|I_1, I_2) &= \int_{\Omega \setminus O} \rho(I_1(x) - I_2(x + f(x)))dx - \alpha \log[Q_{w^*_\varphi, w^*_\psi}(f|I_1)] \\ &= \int_{\Omega \setminus O} \rho(I_1(x) - I_2(x + f(x)))dx + \alpha \|\varphi^*(I_1, \psi^*(f)) - f\|^2 \end{aligned} \quad (7)$$

with $\alpha > 0$, $Q_{w^*_\varphi, w^*_\psi}$ our learned conditional prior, and $\rho(x) = (x^2 + 0.001^2)^\eta$ the generalized Charbonnier penalty function [8]. Note that the integration in the data term is on the co-visible area, i.e. the image domain Ω minus the occluded area O, which can be set to empty for simplicity or modeled using the forward-backward consistency as done in [27] with a penalty on O to prevent trivial solutions. In the following section, we describe our implementation and report results and comparisons on several benchmarks.

3 Experiments

3.1 Network Details

CPN: we adapt the FlowNetS network structure proposed in [15] to be the decoder φ, and the contraction part of FlowNetS to be the encoder ψ in our CPN respectively. Both parts are shrunk versions of the original FlowNetS with a factor of 1/4; altogether our CPN has 2.8M parameters, which is an order of magnitude less than the 38M parameters in FlowNetS. As we mentioned before, the bottleneck in Eq. (6) is controlled by the coding length ℓ_ψ of the encoder ψ, here we make the definition of ℓ_ψ explicit, which is the number of the convolutional kernels in the last layer of the encoder. In our experiments, $\ell_\psi = 128$ always satisfies the stopping criterion described in Sect. 2.3, which ends up with a reduction rate 0.015 in the dimension of the flow f.

CPNFlow: we term our flow prediction network CPNFlow. The network used on all benchmarks for comparison is the original FlowNetS with no modifications, letting us focus on the effects of different loss terms. The total number

of parameters is 38 M. FlowNetS is the most basic network structure for learning optical flow [15], *i.e.*, only convolutional layers with striding for dimension reduction, however, when trained with loss Eq. (7) that contains the learned conditional prior (CPN), it achieves better performance than the more complex network structure FlowNetC [15], or even stack of FlowNetS and FlowNetC. Please refer to Sect. 3.4 for details and quantitative comparisons.

3.2 Datasets for Training

Flying Chairs is a synthesized dataset proposed in [15], by superimposing images of chairs on background images from Flickr. Randomly sampled 2-D affine transformations are applied to both chairs and background images. Thus there are independently moving objects together with background motion. The whole dataset contains about 22 k 512×384 image pairs with ground truth flows.
MPI-Sintel [10] is collected from an animation that made to be realistic. It contains scenes with natural illumination, objects moving fast, and articulated motion. Final and clean versions of the dataset are provided. The final version contains motion blur and fog effects. The training set contains only 1,041 pairs of images, much smaller compared to Flying Chairs.
KITTI 2012 [16] and 2015 [28] are the largest real-world datasets containing ground truth optical flows collected in a driving scenario. The ground truth flows are obtained from simultaneously recorded video and 3-D laser scans, together with some manual corrections. Even though the multi-view extended version contains roughly 15k image pairs, ground truth flows exist for only 394 pairs of image, which makes fully supervised training of optical flow prediction from scratch under this scenario infeasible. However, it provides a base for unsupervised learning of optical flow, and a stage to show the benefit of semi-unsupervised optical flow learning, that utilizes both the conditional prior (CPN) learned from the synthetic dataset, and the virtually unlimited amount of real world videos.

3.3 Training Details

We use Adam [21] as the optimizer with its default parameters in all our experiments. We train our conditional prior network (CPN) using Flying Chairs dataset due to its large amount of synthesized ground truth flows. The initial learning rate is 1.0e-4, and is halved every 100 k steps until the maximum 600 k training steps. The batch size is 8, and the autoencoding loss after training is around 0.6.

There are two versions of our CPNFlow, i.e. CPNFlow-C and CPNFlow-K. Both employ the FlowNetS structure, and they differ in the training set on which Eq. (7) is minimized. CPNFlow-C is trained on Flying Chairs dataset, similarly CPNFlow-K is trained on KITTI dataset with the multi-view extension. The consideration here is: when trained on Flying Chairs dataset, the conditional prior network (CPN) is supposed to only capture the statistics of the affine transformations (a) CPNFlow-C is to test whether our learned prior works properly or not. If it works, (b) CPNFlow-K tests how the learned prior generalizes to

real world scenarios. Both CPNFlow-C and CPNFlow-K have the same training schedule with the initial learning rate 1.0e-4, which is halved every 100 k steps until the maximum 400 k steps.[7] Note that in [33], layer-wise loss adjustment is used during training to simulate coarse-to-fine estimation, however, we will not adopt this training technique to avoid repeatedly interrupting the training process. In a similar spirit, we will not do network stacking as in [19,27], which increases both the training complexity and the network size.

In terms of data augmentation, we apply the same augmentation method as in [15] whenever our network is trained on Flying Chairs dataset with a cropping of 384×448. When trained on KITTI, resized to 384×512, only vertical flipping, horizontal flipping and image order switching are applied. The batch size used for training on Flying Chairs is 8 and on KITTI is 4.

3.4 Benchmark Results

Table 1 summarizes our evaluation on all benchmarks mentioned above, together with quantitative comparisons to the state-of-the-art methods from different categories: Fully supervised, variational, and unsupervised learning methods. Since CPNFlow has the same network structure as FlowNetS, and both CPNFlow-C and FlowNetS are trained on Flying Chairs dataset, the comparison between CPNFlow-C and FlowNetS shows that even if CPNFlow-C is trained without knowing the correspondences between pairs of image and the ground truth flows, it can still achieve similar performance compared to the fully supervised ones on the synthetic dataset MPI-Sintel. When both are applied to KITTI, CPNFlow-C achieves 11.2% and 21.6% improvement over FlowNetS and FlowNetC respectively on KITTI 2012 Train, hence CPNFlow generalizes better to out of domain data.

One might notice that FlowNet2 [19] consistently achieves the highest score on MPI-Sintel and KITTI Train, however, it has a totally different network structure where several FlownetS [15] and FlowNetC [15] are stacked together, and it is trained in a sequential manner, and on additional datasets, e.g. FlyingThings3D [26] and a new dataset designed for small displacement [19], thus not directly comparable to CPNFlow. However, when we simply apply the learned conditional prior to train our CPNFlow on KITTI using Eq. (7), the final network CPNFlow-K surpasses FlowNet2 by 8% on KITTI 2012 Train, yet the training procedure of CPNFlow is much simpler, and there is no need to switch between datasets nor between different modules of the network.

Since the emergence of unsupervised training of optical flow [20], there has not been a single method that beats the variational methods, as shown in Table 1, even if both variational methods and unsupervised learning methods are minimizing the same type of loss function. One reason might be that when we implement the variational methods, we could apply some "secret" operations as mentioned in [34], e.g. median filtering, such that implicit regularization is triggered. Extra data term can also be added to bias the optimization, as in [7],

[7] $\alpha = 0.1, \eta = 0.25$ for CPNFlow-C, and $\alpha = 0.045, \eta = 0.38$ for CPNFlow-K.

Table 1. Quantitative evaluation and comparison to the state-of-the-art optical flow estimation methods coming from three different categories. Sup: Fully supervised, Var: Variational methods, and Unsup: Unsupervised learning methods. The performance measure is the end-point-error (EPE), except for the last column where percentage of erroneous pixels is used. The best performer in each category is highlighted in bold, and the number in parentheses is fine-tuned on the tested dataset. For more detailed comparisons on KITTI test sets, please refer to the online benchmark website: http://www.cvlibs.net/datasets/kitti/eval_flow.php

	Methods	Chairs test	Sintel Train		Sintel Test		KITTI Train		KITTI Test	
			Clean	Final	Clean	Final	2012	2015	2012	2015
Sup	FlowNetS [15]	2.71	4.50	5.45	7.42	8.43	8.26	—	**9.1**	—
	FlowNetC [15]	**2.19**	4.31	5.87	7.28	8.81	9.35	—	—	—
	SPyNet [32]	2.63	4.12	5.57	6.69	8.43	9.12	—	10.1	—
	FlowNet2 [19]	—	**2.02**	**3.14**	**3.96**	**6.02**	**4.09**	**10.06**	—	—
Var	Classic-NL [34]	—	6.03	7.99	7.96	9.15	—	—	16.4	—
	LDOF [7]	**3.47**	**4.29**	6.42	**7.56**	**9.12**	13.7	—	12.4	—
	HornSchunck [35]	—	7.23	8.38	8.73	9.61	—	—	**11.7**	**41.8%**
	DIS-Fast [24]	—	5.61	**6.31**	9.35	10.13	**11.01**	**21.2**	14.4	—
Unsup	DSTFlow [33]	5.11	6.93	7.82	10.40	11.11	16.98	24.30	—	—
	DSTFlow-ft [33]	5.11	(**6.16**)	(**6.81**)	10.41	11.27	10.43	16.79	12.4	**39%**
	BackToBasic [20]	5.30	—	—	—	—	11.30	—	**9.9**	—
	UnFlowC [27]	—	—	—	—	—	7.11	14.17	—	—
	UnFlowC-oc [27]	—	—	8.64	—	—	3.78	8.80	—	—
	UnFlowCSS-oc [27]	—	—	7.91	**9.37**	10.22	**3.29**	**8.10**	—	—
	DenseNetF [40]	**4.73**	—	—	—	**10.07**	—	—	11.6	—
	CPNFlow-C	**3.81**	**4.87**	**5.95**	**7.66**	**8.58**	7.33	14.61	—	—
	CPNFlow-K	4.37	6.46	7.12	—	—	3.76	9.63	4.7	30.8%
	CPNFlow-K-o	—	7.01	7.52	—	—	**3.11**	**7.82**	**3.6**	**30.4%**

sparse matches are used as a data term to deal with large displacements. However, when combined with our learned conditional prior, even the simplest data term would help unsupervisedly train a network that outperforms the state-of-the-art variational optical flow methods. As shown in Table 1 our CPNFlow consistently achieves similar or better performance than LDOF [7], especially on KITTI 2012 Train, the improvement is at least 40%.

Compared to unsupervised optical flow learning, the advantage of our learned conditional prior becomes obvious. Although DenseNetF [40] and UnFlowC [27] employ more powerful network structures than FlowNetS, their EPEs on MPI-Sintel Test are still 1.5 higher than our CPNFlow. Note that in [27], several versions of result are reported, e.g. UnFlowC: trained with brightness data term and second order smoothness term, UnFlowC-oc: census transform based data term together with occlusion modeling and bidirectional flow consistency penalty, and UnFlowCSS-oc: a stack of one FlowNetC and two FlowNetS's sequentially trained using the same loss as in UnFlowC-oc. Our CPNFlow-K outperforms UnFlowC by 47% on KITTI 2012 Train and 32% on KITTI 2015 Train. When occlusion reasoning is effective in Eq. (7) as done in [27], our CPNFlow-K-o outperforms UnFlowC-oc by 17.7% on KITTI 2012 Train, 11.1% on KITTI 2015 Train, and 12.9% on Sintel Train Final, even without a more robust census trans-

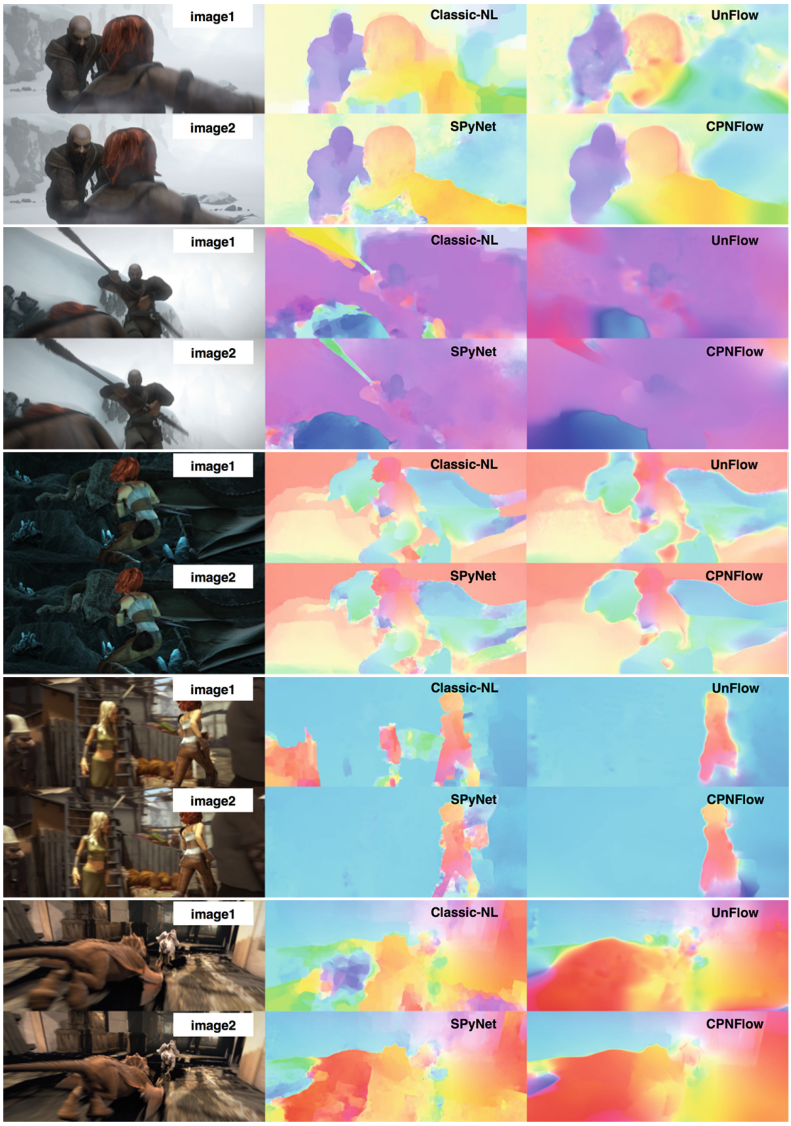

Fig. 4. Visual comparison on MPI-Sintel. Variational: CLassic-NL [34], Supervised: SPyNet [32], Unsupervised: UnFlowC [27] and our CPNFlow-C.

form based data term and flow consistency penalty, which demonstrate the effectiveness of our learned conditional prior across different data terms. Note that our CPNFlow-K-o even outperforms UnFlowCSS-oc, which is far more complex in training and network architecture.

Figures 4, 5, 6 show the visual comparisons on MPI-Sintel, KITTI 2012 and KITTI 2015 respectively. Note that our CPNFlow is generally much smoother,

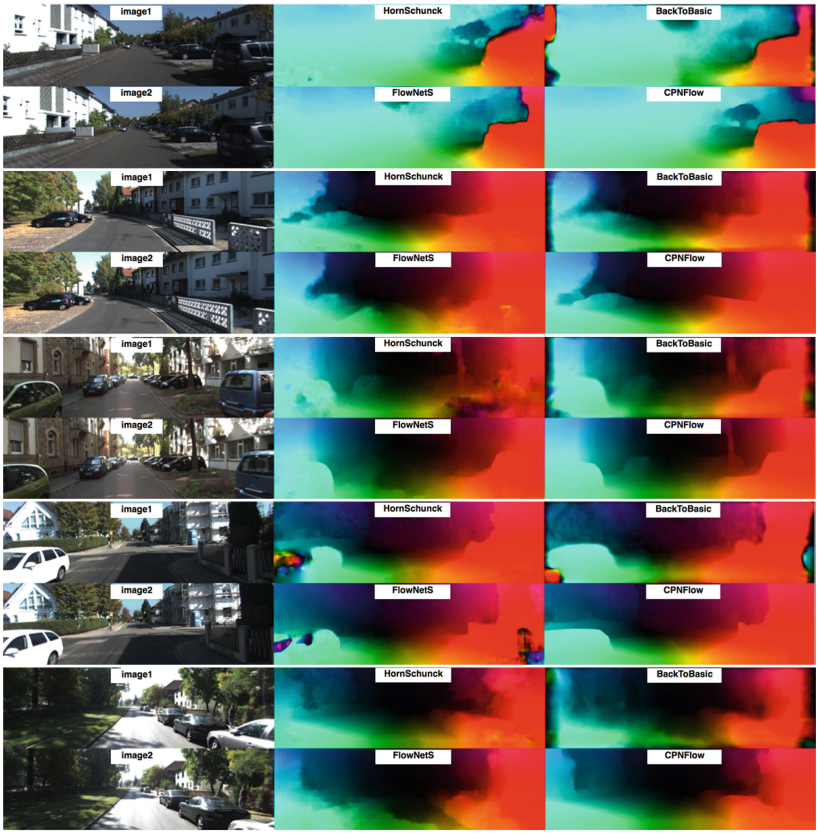

Fig. 5. Visual comparison on KITTI 2012. Variational: HornSchunck [35], Supervised: FlowNetS [15], Unsupervised: BackToBasic [20] and our CPNFlow-K.

and at the same time sharper at object boundaries, e.g. the girl in the 3rd, 4th rows and the dragon in the 5th row in Fig. 4. This demonstrates that our conditional prior network (CPN) is capable of learning high level (semantic) regularities imposed by object entities. In Fig. 5, we can also observe that discontinuities in the flow fields align well with object boundaries, for example the cars in all pairs. This, again, demonstrates that our learned conditional prior is able to generalize to different scenarios. The error of the estimated flows is also displayed in Fig. 6.

4 Discussion and Related Work

Generic priors capturing rudimentary statistics to regularize optical flow have been used for decades, starting with Horn and Schunk's ℓ^2 norm of the gradient, to ℓ^1, Total Variation, etc. We seek to design or learn image-dependent priors that capture long-range correlation and semantics.

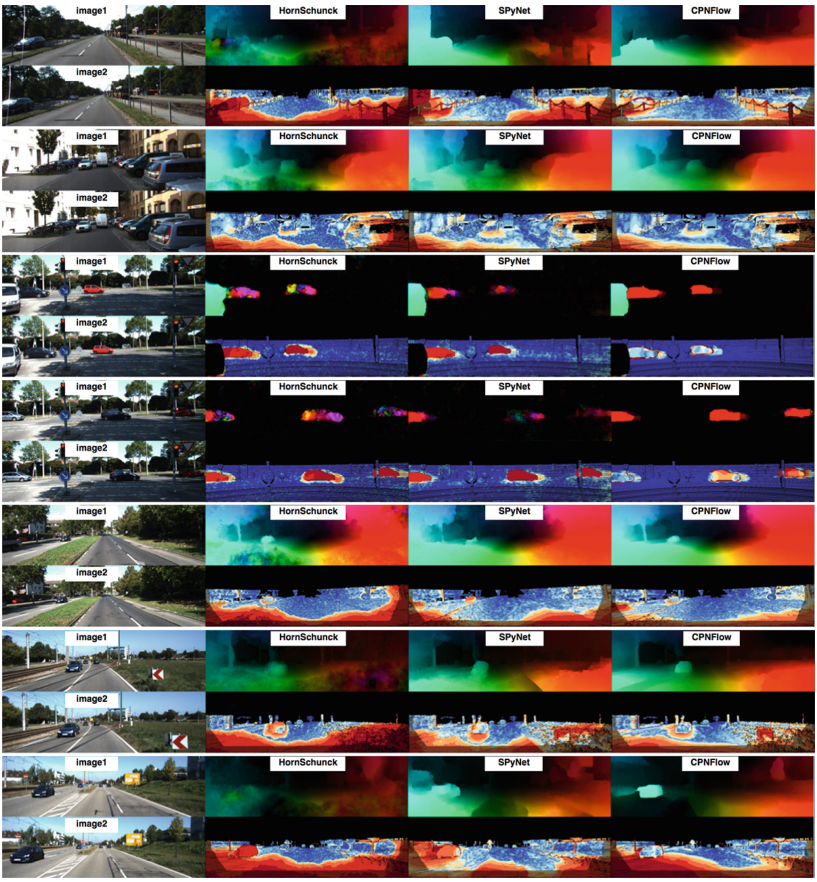

Fig. 6. Visual comparison on KITTI 2015. Variational: HornSchunck [35], Supervised: SPyNet [32] and our CPNFlow-K. The 2nd row in each pair shows the end-point-error of the estimated flow, red is high and blue is low.

Image-dependent priors of the form Eq. (2) include [6,12,14,23,30,31,37], whereas most recent methods learn optical flow end-to-end, without explicitly isolating the likelihood and prior terms, for instance [15,19,32] are the top performing on MPI-Sintel. Some methods even cast optical flow as dense or semi-dense feature matching [3,25,38,39] in order to deal with large displacements, while the regularity is merely imposed by forward-backward matching consistency (see references therein for a detailed review of related literature).

It would be tempting to use a GAN [17] to learn the prior distribution of interest. A GAN can be thought of as a method to learn a map g such that its push-forward g_* maps two distributions, one known μ, and one we can sample from, p, so $\hat{g} = \arg\min \mathbb{KL}(g_*\mu||p)$. It does so via an adversarial process such that a generative model G will capture the data distribution p_{data}. If we sample

from the generative model G, we will have samples that are equivalently sampled from p_{data}, in order to evaluate $p_{data}(x)$ of a sample x, we can not circumvent the sampling step, thus making the method unsuitable for our purpose where we want a differentiable scalar function.

Our work entails constructing an autoencoder of the flow, so it naturally relates to [22]. Similarly, evaluating the probability of a test example is intractable, even if we can approximately evaluate the lower bound of the probability of a data point, which again can not be computed in closed form due to the expectation over the noise.

Optical flow learning algorithms typically rely on synthesized datasets, due to the extreme difficulty in obtaining ground truth flows for realistic videos. Recently, unsupervised optical flow learning methods have flourished, making use of vast amount of unlabeled videos. Although unsupervised optical flow learning methods are able to learn from unlimited amount of data, when compared to variational methods, their performance usually falls behind, even when a similar loss is employed. A phenomenon observed is that almost all unsupervised optical flow learning methods use the Horn-Schunck type surrogate losses. And there is debate on which feature to use for the data term, *e.g.*, the raw photometric value or the edge response, or on the prior/regularizer term, *e.g.* , penalizing the first order gradient of the flow or the second order, or on how to weight the prior term in a pixel-wise manner. Surrogate losses are getting more and more complex. Instead of focusing on the data term, we ask what should be the best form for the prior term. Our answer is that structural consistency between an image and the flow, as well as high-order statistics, such as semantic consistency, are important. We show that when combined with the raw photometric warping error, this kind of prior serves as a better regularizer than all the other hand-designed ones. We show its effectiveness on several contemporary optical flow benchmarks, also thanks to its ability to leverage existing limited supervised (synthetic) datasets and unlimited real world videos.

Acknowledgments. Research supported by ONR N00014-17-1-2072 and ARO W911NF-15-1-0564/66731-CS.

References

1. Achille, A., Soatto, S.: Emergence of invariance and disentangling in deep representations. J. Mach. Learn. Res. (JMLR), in press. (Also in Proceedings of the ICML Workshop on Principled Approaches to Deep Learning; arXiv:1706.01350, May 30, 2017)
2. Ahmadi, A., Patras, I.: Unsupervised convolutional neural networks for motion estimation. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 1629–1633. IEEE (2016)
3. Bailer, C., Taetz, B., Stricker, D.: Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4015–4023 (2015)

4. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. Int. J. Comput. Vis. **92**(1), 1–31 (2011)
5. Black, M.J., Anandan, P.: A framework for the robust estimation of optical flow. In: Proceedings of the Fourth International Conference on Computer Vision, pp. 231–236. IEEE (1993)
6. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24673-2_3
7. Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(3), 500–513 (2011)
8. Bruhn, A., Weickert, J.: Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In: Tenth IEEE International Conference on Computer Vision, ICCV, vol. 1, pp. 749–755. IEEE (2005)
9. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/kanade meets horn/schunck: combining local and global optic flow methods. Int. J. Comput. Vis. **61**(3), 211–231 (2005)
10. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7577, pp. 611–625. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33783-3_44
11. Chaudhari, P., Soatto, S.: Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks (2017). arXiv preprint arXiv:1710.11029
12. Chen, Q., Koltun, V.: Full flow: Optical flow estimation by global optimization over regular grids. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4706–4714 (2016)
13. Cohen, N., Shashua, A.: Inductive bias of deep convolutional networks through pooling geometry (2016). arXiv preprint arXiv:1605.06743
14. Deriche, R., Kornprobst, P., Aubert, G.: Optical-flow estimation while preserving its discontinuities: a variational approach. In: Li, S.Z., Mital, D.P., Teoh, E.K., Wang, H. (eds.) ACCV 1995. LNCS, vol. 1035, pp. 69–80. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-60793-5_63
15. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., et al.: Learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2758–2766 (2015)
16. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3354–3361. IEEE (2012)
17. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al.: Generative adversarial nets. In: Advances in neural information processing systems, pp. 2672–2680 (2014)
18. Horn, B.K., Schunck, B.G.: Determining optical flow. Artif. Intell. **17**(1–3), 185–203 (1981)
19. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2462–2470 (2017)

20. Jason, J.Y., Harley, A.W., Derpanis, K.G.: Back to basics: unsupervised learning of optical flow via brightness constancy and motion smoothness. In: European Conference on Computer Vision, pp. 3–10. Springer (2016)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014). arXiv preprint arXiv:1412.6980
22. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2013). arXiv preprint arXiv:1312.6114
23. Krähenbühl, P., Koltun, V.: Efficient nonlocal regularization for optical flow. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7572, pp. 356–369. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33718-5_26
24. Kroeger, T., Timofte, R., Dai, D., Van Gool, L.: Fast optical flow using dense inverse search. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 471–488. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_29
25. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: SIFT flow: dense correspondence across different scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5304, pp. 28–42. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88690-7_3
26. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., et al.: 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, pp. 4040–4048 (2016)
27. Meister, S., Hur, J., Roth, S.: UnFlow: unsupervised learning of optical flow with a bidirectional census loss. In: AAAI. New Orleans, Louisiana (Feb 2018)
28. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
29. Papenberg, N., Bruhn, A., Brox, T., Didas, S., Weickert, J.: Highly accurate optic flow computation with theoretically justified warping. Int. J. Comput. Vis. **67**(2), 141–158 (2006)
30. Proesmans, M., Van Gool, L., Pauwels, E., Oosterlinck, A.: Determination of optical flow and its discontinuities using non-linear diffusion. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 294–304. Springer, Heidelberg (1994). https://doi.org/10.1007/BFb0028362
31. Ranftl, R., Bredies, K., Pock, T.: Non-local total generalized variation for optical flow estimation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 439–454. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_29
32. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4161–4170 (2017)
33. Ren, Z., Yan, J., Ni, B., Liu, B., Yang, X., Zha, H.: Unsupervised deep learning for optical flow estimation. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
34. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2432–2439. IEEE (2010)
35. Sun, D., Roth, S., Black, M.J.: A quantitative analysis of current practices in optical flow estimation and the principles behind them. Int. J. Comput. Vis. **106**(2), 115–137 (2014)

36. Sutton, C., McCallum, A.: An introduction to conditional random fields. Found. Trends® Mach. Learn. **4**(4), 267–373 (2012)
37. Xu, L., Jia, J., Matsushita, Y.: Motion detail preserving optical flow estimation. IEEE Trans. Pattern Anal. Mach. Intell. **34**(9), 1744–1757 (2012)
38. Yang, Y., Lu, Z., Sundaramoorthi, G., et al.: Coarse-to-fine region selection and matching. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5051–5059. IEEE (2015)
39. Yang, Y., Soatto, S.: S2f: Slow-to-fast interpolator flow. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3767–3776. IEEE (2017)
40. Zhu, Y., Newsam, S.: Densenet for dense flow (2017). arXiv preprint arXiv:1707.06316