Unsupervised Deep Learning for Optical Flow Estimation

Zhe Ren,[1] **Junchi Yan,**[2,3*] **Bingbing Ni,**[1] **Bin Liu,**[4] **Xiaokang Yang,**[1] **Hongyuan Zha**[5]
[1]Shanghai Jiao Tong University [2]East China Normal University [3]IBM Research [4]Moshanghua Tech [5]Georgia Tech
{sunshinezhe,nibingbing,xkyang}@sjtu.edu.cn, {jcyan,zha}@sei.ecnu.edu.cn, liubin@dress-plus.com, zha@cc.gatech.edu

Abstract

Recent work has shown that optical flow estimation can be formulated as a supervised learning problem. Moreover, convolutional networks have been successfully applied to this task. However, supervised flow learning is obfuscated by the shortage of labeled training data. As a consequence, existing methods have to turn to large synthetic datasets for easily computer generated ground truth. In this work, we explore if a deep network for flow estimation can be trained without supervision. Using image warping by the estimated flow, we devise a simple yet effective unsupervised method for learning optical flow, by directly minimizing photometric consistency. We demonstrate that a flow network can be trained from end-to-end using our unsupervised scheme. In some cases, our results come tantalizingly close to the performance of methods trained with full supervision.

Introduction

Massive amounts of digital videos are generated every minute. This has posed new challenges for effective video analytics. Estimating pixel-level motions, also known as optical flow, is a basic building block for early-stage video analysis. Optical flow is a classic problem in computer vision and has many real-world applications, including autonomous driving, video segmentation and video semantic understanding (Menze and Geiger 2015). However, accurate estimation of optical flow remains a challenging problem (Sun, Roth, and Black 2014; Butler et al. 2012).

Deep learning has drastically advanced all frontiers of AI, in particular computer vision. We have witnessed a cornucopia of Convolutional Neural Networks (CNN) achieving superior performance in a large array of computer vision tasks, including image denoising, image segmentation and object recognition. Several recent advances also allow for pixel-wise predictions like semantic segmentation (Long, Shelhamer, and Darrell 2015) and trajectory analysis (Lin et al. 2017). However, the ravenous appetite to labeled data becomes the main limitation of deep learning methods. This is even pronounced for the problem addressed in this paper: optical flow estimation that needs dense labels with per-pixel motion between two consecutive frames. Getting such optical flow ground-truth for realistic videos is extremely challenging (Butler et al. 2012). Hence state-of-the-art deep learning methods (Fischer et al. 2015; Mayer et al. 2016) turn to synthetically labeled dataset, bypassing the tedious and difficult pixel-level labeling step. A crowd-sourcing based study (Altwaijry et al. 2016) shows that human participants are mainly relying on the global appearance cues for perceiving motion and human are less attentive to the fine-grained pixel-level correspondences.

Is pixel-level supervision indispensable for learning optical flow? Recent work on learning from video has shown that via some quality control, effective feature representation (Wang and Gupta 2015; Li et al. 2016) and even cross-instance key-point matching (Zhou et al. 2016) can be obtained by unsupervised or semi-supervised learning. Another observation is that the human brain bears a visual short-term memory (VSTM) module (Hollingworth 2004), which is mainly responsible for understanding visual changes, and an infant without any teaching by the age of 2.5 months is able to discern occlusion, containment, and

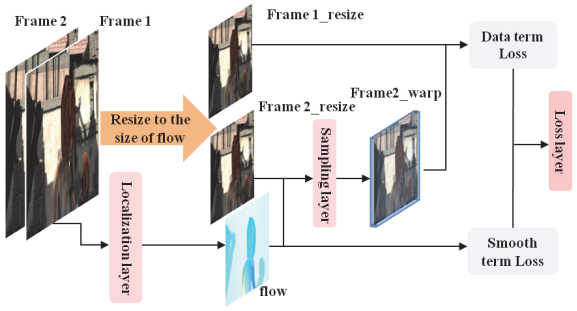


Figure 1: The presented network architectures of our Dense Spatial Transform Flow (DSTFlow) network that consists of three key components: localization layer based on a similar structure of flowNet, sampling layer based on dense spatial transform which is realized by a bilinear interpolation layer in this paper and the final loss layer. All the layer weights are learned end-to-end through backpropagation.

*Correspondence author. This research was supported by The National Key Research and Development Program of China (2016YFB1001003), NSFC (61602176, 61672231, 61527804, 61521062), STCSM (15JC1401700, 14XD1402100), China Postdoctoral Science Foundation Funded Project (2016M590337), the 111 Program (B07022) and NSF (IIS-1639792, DMS-1620345).

covering events which requires motion understanding (Baillargeon 2004). These computational and biological studies suggest that we can potentially learn optical flow without (or with very little) ground-truth labels. In this work, we explore whether a neural network can be trained entirely based on photometric consistency and without supervision.

Given two input images, our network starts with a localization net using the FlowNet structure (Fischer et al. 2015). Localization net outputs the pixel-level translation estimation. This is fed into a bilinear sampling net to generate the spatially warping feature map. Finally the photometric error between the warped feature map from the source image and the target image is taken as the loss, measured by the objective function widely used in learning-free variational methods (Brox et al. 2004). Our network can be regarded as a pixel-level embodiment tailored for optical flow of the recently proposed Spatial Transformer Network (STN) (Jaderberg et al. 2015), which is originally designed for object level transformation modeling. Another fundamental difference is that STN learns the transformation in the context of recognition using object level supervision, while our method utilizes a loss in a fully unsupervised setting.

The main result of our paper is that the deep network trained using our unsupervised scheme, approaches the level of performance of fully supervised training. We believe that this is largely due to our end-to-end training, which allows the network to leverage context information within a large region for inferring local motion. To this end, we summarize our main contributions as follows.

1) To our best knowledge, this is one of the first works for learning optical flow using a deep neural network without any supervision. Our work is fundamentally different from the state-of-the-art learning-free methods DeepFlow (Weinzaepfel et al. 2013) or EpicFlow (Revaud et al. 2015), and the supervised deep learning approach FlowNet (Fischer et al. 2015) and DispNet (Mayer et al. 2016).

2) We propose a novel optical flow network which can be seen akin to the pipeline of Spatial Transformer Network (Jaderberg et al. 2015), leveraging the loss function used in variational methods (Brox et al. 2004) without supervision, for end-to-end unsupervised learning for optical flow estimation. While the gains are modest, we believe this is a promising direction for future exploration.

3) Finally, to enable comparison and further innovation, we will provide a public Caffe (Jia et al. 2014) implementation of our method after the release of this paper.

Related work

We address the problem of unsupervised learning of optical flow using a convolutional network from videos. To begin with, we provide a brief survey of recent methods on optical flow estimation and learning from video. Our method is inspired by the spatial transformer network (Jaderberg et al. 2015), which are discussed in the last part of this section.

Optical flow Optical flow is a classic problem in computer vision. Despite the abundant literature on the topic, it is still very challenging (Sun, Roth, and Black 2014). Many optical flow methods are based on variational method which formulated as an energy minimization problem (Horn and Schunck 1981). This paradigm relies on the photometric consistency of color and gradient, as well as spatial smoothness of natural images. While being attractive, such methods can get stuck in local minima with error accumulation across scales, and tend to fail against large displacements.

To tackle this issue, Large Displacement Optical Flow (LDOF) (Brox and Malik 2011) combines local descriptor matching with the variational method. Local descriptors e.g. HOG (Mikolajczyk and Schmid 2005) are extracted in rigid local frames and matched across the images. These matching results are used to initialize flow estimation, which are further optimized using variational method within a coarse-to-fine pyramid. Xu et. al (Xu, Jia, and Matsushita 2012) further integrates more advanced matching methods such as SIFT-flow (Liu, Yuen, and Torralba 2011) and PatchMatch (Barnes et al. 2010) to increase the accuracy of flow.

HOG-like features are recently replaced by a CNN-inspired patch matching scheme: DeepMatching (Revaud et al. 2016), leading to state-of-the-art DeepFlow method (Weinzaepfel et al. 2013). DeepFlow follows a fine-to-coarse procedure. It starts with local patch matching, builds a progressively lower resolution matching map via max-pooling (LeCun et al. 1998), and followed by the traditional energy minimization framework. The successive work EpicFlow (Revaud et al. 2015) improves DeepFlow by leveraging contour cues to constrain the flow map. This is done by a sparse-to-dense interpolation from an initial set of matches, where the weights are defined by edge-aware geodesic distance. Both DeepFlow and EpicFlow are learning-free in the sense that features are hand-crafted and no learning is involved.

While optical flow estimation is well-established, little has been done using end-to-end convolutional network until recent work (Fischer et al. 2015). Rather than rely on hand-designed features, FlowNet (Fischer et al. 2015) poses optical flow as a supervised learning task and utilize an end-to-end convolutional network to predict the flow field. DispNet (Mayer et al. 2016) extends this idea to disparity and scene flow estimation. However, these neural network methods require strong supervision for training. Specifically, training of both the FlowNet and DispNet is enabled by large synthetically generated datasets. In this work, we explore whether unsupervised learning can be used instead.

Finally we emphasize learning to match with CNN is not a brand-new idea (Zbontar and LeCun 2015; Zagoruyko and Komodakis 2015; Fischer et al. 2015). Unlike learning to match local patches (Zbontar and LeCun 2015; Zagoruyko and Komodakis 2015), our method directly predict the pixel-level offsets of an input frame pair. While our network is similar to (Fischer et al. 2015), our training is completely unsupervised. To our best knowledge, the most relevant work is (Zhou et al. 2016). Our method differs from (Zhou et al. 2016) in three aspects: i) they leverage additional 3-D CAD model to establish the cross-instance correspondence chain, and as a result the training images need to be similar or in the same category as the available CAD models; ii) they use a cycle-consistency loss which need to involves four images one time; iii) they mainly address the cross-instance match-

ing problem and do not focus on the optical flow scenario which usually involve consecutive video frames. Therefore, our contribution is orthogonal to these previous work.

Learning from video There is an emerging interest for learning visual representations from video *itself*, in a semi-supervised or unsupervised manner. Seminal work (van Hateren and Ruderman 1998; Hurri and Hyvarinen 2003) are based on Independent Component Analysis (ICA) using the concept of temporal coherence. Srivastava et. al (Srivastava, Mansimov, and Salakhutdinov 2015) use the Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) Encoder-Decoder framework to learn video representations. Goroshi et. al (R. Goroshin and LeCun 2015) propose to perform feature learning from videos by imposing the constraint that temporally close frames should have similar feature representations. Moreover, Wang and Gupta (Wang and Gupta 2015) learn general-purpose CNN feature by enforcing that the tracked patches in different frames should have a similar visual representation. (Agrawal, Carreira, and Malik 2015) exploits the awareness of ego-motion as supervision for feature learning, whereby the global transformation regarding the ego-motion is assumed given by camera pose. However, while in previous approaches the training objective was used as a surrogate to encourage the network to learn a useful representation, our primary goal is to train an optical flow model at pixel-level, and the learned representation associated with the network is simply a useful byproduct. In this spirit, our work is akin to the idea of learning edge detector from video (Li et al. 2016). However, the designed network and objective are both very different as edge detection once involves only a single image while optical flow need to match pixels across a pair of frames.

Learning by disentangling pose and identity Optical flow seeks to estimate the motion of pixels or local patches. However, vanilla CNNs only have limited predefined pooling mechanism to handle spatial variations. To enable more flexible spatial transformation capabilities, (Hinton, Krizhevsky, and Wang 2011) learn a hierarchy of units that locally transform their input for generating small rotations to an input stereo pair. (Cheung et al. 2014) propose an auto-encoder with decoupled semantic units representing pose and identity. More recently, Spatial Transformer Network (STN) is proposed by (Jaderberg et al. 2015) as an attention mechanism (Mnih et al. 2014; Ba, Mnih, and Kavukcuoglu 2015) capable of warping the inputs to increase recognition accuracy. It takes as input the image and produces the parameters for a pre-determined transformation model which is used in turn to transform the image. (Altwaijry et al. 2016) uses STN for image similarity verification as the model can be trained with standard back-propagation, unlike the attention mechanisms of (Mnih et al. 2014; Ba, Mnih, and Kavukcuoglu 2015) that relied on reinforcement learning techniques. Our method is inspired by the paradigm of STN and injects a sampling layer to the network which allows for end-to-end backpropagation.

Network and Training Procedure

Using a pair of images as input, our optical flow network can be conceptually regarded as a Spatial Transformation Network (Jaderberg et al. 2015) with three major components: i) the localization part which outputs the pixel-level translation i.e. flow as shown in Fig.1 via a FlowNet structure; ii) a bilinear sampling net for generating the warped frame via the translation output by the localization net; iii) the final layer involving a loss function regarding the discrepancy between the target image and the warped one from the source image. The loss is similar to the classic objective function used in previous work (Brox et al. 2004) based on the variational framework, but we use it to train the flow network weights rather than estimating the flow map. We term our network as Dense Spatial Transform Flow (DSTFlow) (see Fig.1 for an overview), which is elaborated as follows:

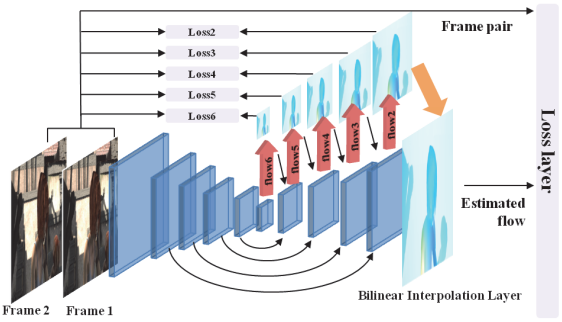


Figure 2: Information flow of our neural network for unsupervised optical flow learning. The convolutional layers in the middle are similar to the the simplified version of FlowNet (see Fig.2 in (Fischer et al. 2015)). As a result, similar multi-scale loss summation is used to train the network.

Localization net We adopt the network structure from FlowNet, and specifically the FlowNetSimple (see Fig.2 in (Fischer et al. 2015)) as the network for computing the pixel-level offset. FlowNet is so-far the few network allowing for end-to-end flow map estimation i.e. pixel offset computing and the model is differentially trainable by backpropagation. One potential alternative is the network presented in Fig.2 of the work (Zhou et al. 2016) which was intended for cross-instance key point matching. We leave it for future work.

The localization layers take a pair of images as input, and output the x-y flow fields. For simplicity and generality, we stack both input images together with $3 \times 2 = 6$ channels, which we borrow from the FlowNetSimple (Fischer et al. 2015). As shown in Fig.2, it is comprised by 10 convolutional layers followed by a refinement whose main ingredient is 'upconvolutional' a.k.a. 'deconvolutional' layers (Zeiler and Fergus 2014), which can be seen as an unpooling (extending the feature maps, as opposed to pooling) and a convolution operations. We concatenate the 'deconvolutional' output with the feature maps from the corresponding 'conv' layer and an unpooling flow prediction from previous scale. By doing so, both the high-level and fine-grained information for the prediction layer can be well maintained.

Sampling net The sampling layer helps warp the input feature map to a transformed output map using the output transformation from the localization net. Specifically, each (x_i^s, y_i^s) coordinate defines the spatial location in the input feature map U where a sampling kernel is applied to get

the value at a particular pixel i in the output feature map V. While U^c_{nm} is the value at location (m, n) of the input feature map along channel c, and V^c_i is the output one for pixel i. To allow differentiable stochastic (sub)-gradient descent, we adopt a bilinear sampling kernel:

$$V^c_i = \sum_{n=1}^{H}\sum_{m=1}^{W} U^c_{nm}\max(0, 1-|x^s_i - m|)\max(0, 1-|y^s_i - n|)$$

Backpropagation is done by computing partial derivatives:

$$\frac{\partial V^c_i}{\partial U^c_{nm}} = \sum_{n=1}^{H}\sum_{m=1}^{W} \max(0, 1-|x^s_i - m|)\max(0, 1-|y^s_i - n|)$$

$$\frac{\partial V^c_i}{\partial x^s_i} = \sum_{n=1}^{H}\sum_{m=1}^{W} U^c_{nm}\max(0, 1-|y^s_i - n|) \begin{cases} 0 & \text{if } |m - x^s_i| \geq 1 \\ 1 & \text{if } m \geq x^s_i \\ -1 & \text{if } m < x^s_i \end{cases}$$

where $\frac{\partial V^c_i}{\partial y^s_i}$ can be computed in a similar fashion.

Note that when we get the flow2 (see the right most brown arrow in Fig.2) whose size is one fourth of the original frame, we neither continue to repeat the deconv operation nor just simply enlarge the size by traditional bilinear interpolation. Instead, we make an enlargement through spatial bilinear transform which is a special case of our dense spatial transform, not only saving cost both on time and space via simplifying the network, but also incorporating enlargement into the end-to-end training of whole network. In another word, we can seamlessly integrate the traditional bilinear interpolation into our network pipeline.

Loss layer We follow the traditional loss function used in learning-free variational methods (Brox et al. 2004; Brox and Malik 2011) which resemble the original formulation of (Horn and Schunck 1981). This is also motivated by the very recent work (Chen and Koltun 2016) which shows that such flow objective itself is sufficiently powerful to produce accurate mappings even in the presence of large displacements. It combines a data term that assumes constancy over time of some image property and a spatial (and often smooth) term modeling how the flow is expected to vary across image. The data loss measures the discrepancy between one input image and the warped image from the other image, based on the predicted optical flow field. The smooth term models the difference among the neighboring flow predictions. Here we could interpret the loss as a surrogate relating to the desired properties of the ground-truth, i.e. while we do not know what the ground-truth is, we know how it should behave.

For image I_1 and I_2, as a common core feature of most optical flow algorithms, we choose to model grey and gradient constancy by a Charbonnier penalty (Bruhn and Weickert 2005) $\Psi(s) = \sqrt{(s^2 + 0.001^2)}$: a differentiable variant of the absolute value, and a robust convex function against outliers and noises. It is also recommended by recent systematic evaluation in (Sun, Roth, and Black 2014).

$$\ell_D = \int_\Omega \Psi(|I_2(x+w) - I_1(x)|^2 + \gamma|\nabla I_2(x+w) - \nabla I_1(x)|^2)dx \quad (1)$$

The smooth term is modeled by:

$$\ell_S = \int_\Omega \Psi(|\nabla u(x)|^2 + |\nabla v(x)|^2)dx \quad (2)$$

Table 1: Overview of datasets. GT denotes Ground Truth. 'Flying Chairs' is the synthetically generated dataset. KITTI has two versions. Note in KITTI, single-view (s-view) samples are in pair and labeled with ground truth, and the multi-view (m-view) is an extension set without ground truth. Note 'Sintel Clean' and 'Sintel Final' have the same size.

Dataset	Flying chair	KITTI Benchmark Suite 2012		KITTI Benchmark Suite 2015		Sintel
		s-view	m-view	s-view	m-view	
#frames	45744	778	8124	800	8400	1628
#pairs	22872	389	7736	400	8000	1593
#train pairs	22232	194	—	200	—	1041
#test pairs	640	195	—	200	—	552
has GT?	yes	sparse	no	sparse	no	yes

Incorporating the above terms, the overall loss as termed by Dense Spatial Transform (DST) loss is[1]:

$$\ell_{dst} = \ell_D + \alpha\ell_S \quad (3)$$

Multi-scale loss accumulation FlowNet can output flow in multi-scale which improve the predicted flow layer by layer using multi-scale ground truth. Similarly, we impose our DST loss on multi-scale input frame after downsampling. As shown in Fig.2, there are in total six loss layers. In summary, we use the summed loss to guide the information flow over the network. The parameters are updated via standard backpropagation since all parts are differentially trainable.

One shall note that our loss is dramatically different from the endpoint error (EPE) loss used in the supervised learning FlowNet, i.e. the Euclidean distance between the predicted flow vector and the ground truth, averaged over all pixels.

Training Akin to the Flownet, we perform data augmentation to avoid overfitting by imposing mirror, translate, rotate, scaling (spatial augmentation), and contrast, gamma and brightness transformation (chromatic augmentation).

We use Rectified Linear Units (ReLU) for all our nonlinearities, and train the networks with Stochastic Gradient Descent. To handle the six multi-scale loss layers, we adopt loss weight schedule (Mayer et al. 2016) which gradually trains the network from bottom loss layers to top ones, until adding them up for further training.

Finally we shall point out that although our network consist of three parts and seems more complicated than FlowNet, there will be only localization net involved during test stage, which runs as fast the same as the FlowNet.

Experiments and Discussion

Datasets and peer methods

We evaluate our method on three modern datasets:

MPI-Sintel dataset (Butler et al. 2012) is obtained from an animated movie which pays special attention to realistic image effects. It contains multiple sequences including

[1]One can further add the matching term as used in (Brox and Malik 2011; Revaud et al. 2016): $\ell_M = \int_\Omega \Psi(|f_1(x) - f_2(x)|^2)dx$ where $f_1(x)$ and $f_2(x)$ denote for feature vector extracted by certain means such as Sift and CNN. We leave this for future work.

Table 2: Average endpoint errors i.e. EPE (in pixels) over occluded (OCC) and non-occluded areas (NOC) of our networks compared to peer methods and the variation of our method on different datasets. DSTFlow(C+K) denotes train DSTFlow using 'Chairs' first, and then refine with 'KITTI'. DSTFlow(C+S) denotes train DSTFlow using 'Chairs' first, and then refine with 'Sintel'. f1-all:percentage of optical flow outlier over all the pixel. It counts the point correct only if the end-to-end error of this point is < 3px or $< 5\%$ compared with the ground truth. Note the numbers with asterisk are the results of the networks on data they were trained on, and hence are not directly comparable to other results as they tend to overfitting.

Method	Chairs	KITTI2012				KITTI2015				Sintel Clean				Sintel Final				Time (Sec)	
	testing	train		test		train			test	train		test		train		test		CPU	GPU
		occ	noc	occ	noc	occ	noc	fl-all	fl-all	occ	noc	occ	noc	occ	noc	occ	noc		
Epicflow	2.94	3.47	1.48	3.8	1.5	9.57	4.45	0.28	0.27	2.40	1.23	4.12	1.36	3.70	2.63	6.29	3.06	16	—
Deepflow	3.53	4.58	2.22	5.8	1.5	13.89	6.75	0.31	0.30	3.31	1.78	5.38	1.77	4.56	3.07	7.21	3.34	17	—
EPPM	—	—	—	9.2	2.5	—	—	—	—	—	—	6.49	2.68	—	—	8.38	4.29	—	0.2
LDOF	3.47	13.73	4.62	12.4	5.6	18.23	9.05	0.37	—	4.29	2.83	7.56	3.43	6.42	4.41	9.12	5.04	65	2.5
FlowNetS(C+S)	3.04	7.52	4.25	9.1	5.0	14.19	8.12	0.51	—	*3.66	*2.82	6.96	—	*4.44	*3.99	7.76	—	—	0.08
DSTFlow(Chairs)	5.11	16.98	9.21	—	—	24.30	14.23	0.52	—	6.93	5.05	10.40	5.20	7.82	5.97	11.11	5.92	—	0.08
DSTFlow(KITTI)	6.86	10.43	3.29	12.4	4.0	16.79	6.96	0.36	0.39	7.10	5.26	10.95	5.87	7.95	6.16	11.80	6.70	—	0.08
DSTFlow(Sintel)	5.68	15.78	8.24	—	—	23.69	13.88	0.55	—	*6.16	*4.17	10.41	5.30	*7.38	*5.45	11.28	6.16	—	0.08
DSTFlow(C+K)	5.86	16.17	8.32	—	—	22.93	12.81	0.48	—	7.51	5.74	—	—	8.29	6.55	—	—	—	0.08
DSTFlow(C+S)	5.52	17.17	9.52	—	—	25.98	15.89	0.53	—	*6.47	*4.61	10.84	5.62	*6.81	*4.91	11.27	6.02	—	0.08

large/rapid motions. We use both the 'clean' and 'final' version images to train the model.

KITTI dataset (Geiger, Lenz, and Urtasun 2012) contains photos shot in city streets from a driving platform. It offers greater challenges regarding with large displacements, different materials, a large variety of lighting conditions. 'KITTI2012' (Geiger et al. 2013) consists of 194 training pairs and 195 test pairs while KITTI2015 (Menze and Geiger 2015) consists of 200 training pairs and 200 test pairs. In both datasets, there are also multi-view extension datasets which is 20 frames per scene but with no ground truth provided. In our experiment, we make the multi-view extended versions (without ground truth) of the two datasets together as the training dataset with 13372 image pairs, and use pairs with ground truth as our validation set with 194 pairs for 'KITTI2012' and 200 for 'KITTI2015' respectively. Finally we test our model online using the testing protocol from KITTI website[2]. Note that we have excluded the pairs with ground truth and their neighboring two frames in multi-view datasets for unsupervised training to avoid the mixture of training and testing samples.

Flying Chairs The dataset (Fischer et al. 2015) is a recently released synthetic benchmark which consists of segmented background images from Flickr on which the random images of chairs from (Aubry et al. 2014) are overlayed. Flownet (Fischer et al. 2015) has shown that it can be used to train a model as supervision, although it is created artificially. As the same with (Fischer et al. 2015) we split the dataset into 22232 samples (i.e. image pairs) for training and 640 samples for testing, respectively.

An overview of used datasets is given in Table 1. More details for the datasets can be found in (Fischer et al. 2015).

We compare state-of-the-art methods including EpicFlow (Revaud et al. 2015), EPPM (Bao, Yang, and Jin 2014), DeepFlow (Weinzaepfel et al. 2013), LDOF (Brox and Malik 2011), FlowNet (Fischer et al. 2015). For the supervised learning method FlowNet, we use the model publicly available on website which is termed as FlowNet+ft in (Fischer et al. 2015). While here, we term it as FlowNet(C+S) which is trained on 'Chairs' dataset and further finetuned on 'Sintel'. Note other peer methods are all learning-free.

[2]http://www.cvlibs.net/datasets/kitti/user_login.php

(a) Raw image (b) Ground truth (c) DSTFlow

(d) FlowNet (e) DeepFlow (f) EpicFlow

Figure 3: Examples of flow prediction on Chairs.

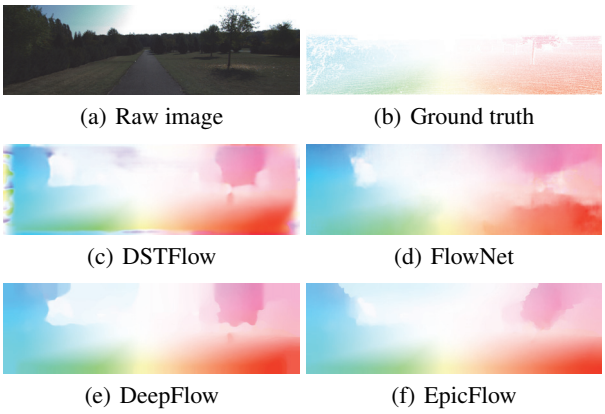
Training and evaluation protocol

For loss function, we set $\alpha = 2$ in Eq.3 and $\gamma = 1$ in Eq.1. In line with (Fischer et al. 2015), we adopt the Adam method and set its parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The start learning rate λ is set by $1e-4$ which decreases half for every 6000 iterations after first 30000 iterations. The batch size is set to 64. For finetune, we start with a learning rate $1e-5$.

Note that since 'Flying Chairs' has abundant samples, we tentatively use it to initialize our model from scratch using our unsupervised training procedure, and then finetune the model using the 'KITTI' and 'MPI-Sintel' data for performance evaluation on these two datasets respectively.

Results and discussion

The evaluation results regarding with the endpoint error (EPE) on the training and testing sets are reported in Table 2. Visual results are presented in Fig.3, Fig.4, Fig.5 for the dataset 'Flying Chairs', 'KITTI2012' and 'Sintel' respec-

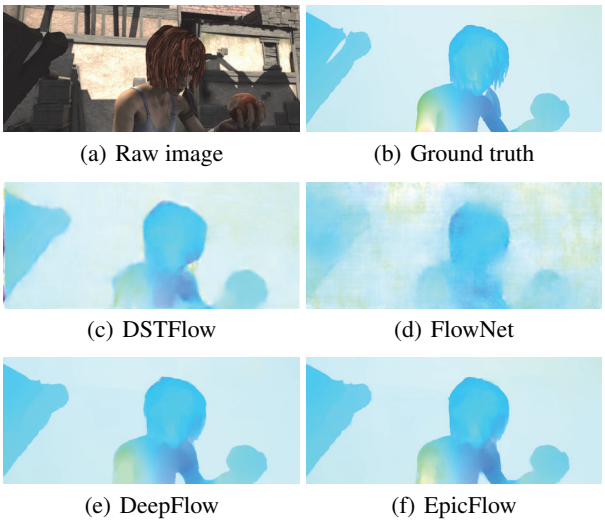


(a) Raw image (b) Ground truth

(c) DSTFlow (d) FlowNet

(e) DeepFlow (f) EpicFlow

Figure 4: Examples of flow prediction on KITTI2012.



(a) Raw image (b) Ground truth

(c) DSTFlow (d) FlowNet

(e) DeepFlow (f) EpicFlow

Figure 5: Examples of flow prediction on Sintel.

tively. Our method DSTFlow is trained on the three datasets respectively. To visualize the flow fields, we use the tool provided by Sintel (Butler et al. 2012). Flow direction is encoded with color and magnitude with color intensity.

Based on the quantitative results in Table 2, we present our analysis for each variant of our method respectively:

DSTFlow(Chairs) It means the model is trained on the Flying Chairs dataset by unsupervised learning. DSTFlow(Chairs) achieves a reasonable performance on the test set. Note that on the very challenging benchmark 'Sintel Final', it performs very close to LDOF (Brox and Malik 2011), and on the 'KITTI2015' training set, it also achieves similar performance compared with FlowNet on the metric f1-all. These results are encouraging especially considering we only use computer generated data for unsupervised learning.

DSTFlow(KITTI) and DSTFlow(C+K) We also train our model on the 'KITTI' data. The label information is very sparse on KITTI and a very small ratio of samples have label information. Unsupervised learning is favored in such cases.

Specifically, we test two variants. DSTFlow(KITTI) denotes we use the filtered multi-view (without ground truth) data that consists of 13372 image pairs as the training data for unsupervised learning. Though the training set size is smaller than 'Flying Chairs', it is significantly larger than the hundreds of labeled image pairs in the KITTI dataset. In contrast to DSTFlow(Chairs), it is trained on natural images. The other version DSTFlow(C+K) is the model first trained on the 'Flying Chairs' data and then refined by the 'KITTI' data, all in an unsupervised setting. We observe that in the finetuning stage, it takes less iterations to get reasonable objective. It is also notable that DSTFlow(KITTI) outperforms the learning-free method LDOF on both 'KITTI' 2012 and 2015, and even slightly better than the supervised learning method FlowNet on the test set of 'KITTI2012' for the 'non-occluded' case. On the training set of 'KITTI2015', DSTFlow(KITTI) also performs competitively against its supervised counterpart FlowNet.

DSTFLOW(Sintel) and **DSTFlow(C+S)** Although Sintel dataset only provides 1041 pairs of images for training, we still train a model from scratch on sintel by imposing extensive data augmentation which is termed as DSTFLOW(Sintel). DSTFlow(C+S) denotes training the model using the 'Flying Chairs' data and finetuning the model using the 'Sintel' data. Though both models perform better on the training set, they both degrade on the testing set compared with DSTFlow(Chairs).

Potential on larger dataset without labeling Though at present, our unsupervised model does not surpass state-of-the-art supervised learning based models, we believe there is some space to explore this possibility. The behind logic is that currently the training datasets even for unsupervised learning still have very limited size, e.g. even for the synthetic dataset 'Flying Chairs', the number of image pairs is less than 23000. Such status quo might not give full potential to the unsupervised flow network. In this spirit, we conjecture the unsupervised optical flow network can be further improved given more training data without supervision.

Conclusion and Outlook

We have presented an end-to-end differentiable optical flow network trained in a unsupervised fashion, which to our knowledge is the first network for unsupervised optical flow learning. Though its current performance slightly lags behind state-of-the-arts, we believe it is a promising direction due to the possibility of leveraging massive readily available video data , and new loss function under the proposed framework. One immediate future work is integrating the matching term in (Brox and Malik 2011) for loss function.

References

Agrawal, P.; Carreira, J.; and Malik, J. 2015. Unsupervised learning of visual representations using videos. In *ICCV*.

Altwaijry, H.; Trulls, E.; Hays, J.; Fua, P.; and Belongie, S. 2016. Learning to match aerial images with deep attentive architectures. In *CVPR*.

Aubry, M.; Maturana, D.; Efros, A.; Russell, B.; and Sivic, J. 2014. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*.

Ba, J.; Mnih, V.; and Kavukcuoglu, K. 2015. Multiple object recognition with visual attention. In *ICLR*.

Baillargeon, R. 2004. Infants' physical world. *American Psychological Society* 13(3).

Bao, L.; Yang, Q.; and Jin, H. 2014. Fast edge-preserving patchmatch for large displacement optical flow. In *CVPR*.

Barnes, C.; Shechtman, E.; Goldman, D.; and Finkelstein, A. 2010. The generalized patchmatch correspondence algorithm. In *ECCV*.

Brox, T., and Malik, J. 2011. Large displacement optical flow: descriptor matching in variational motion estimation. *TPAMI* 33(3):500–513.

Brox, T.; Bruhn, A.; Papenberg, N.; and Weickert, J. 2004. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 25–36.

Bruhn, A., and Weickert, J. 2005. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *ICCV*.

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

Chen, Q., and Koltun, V. 2016. Full flow: Optical flow estimation by global optimization over regular grids. In *CVPR*.

Cheung, B.; Livezey, J. A.; Bansal, A. K.; and Olshausen, B. A. 2014. Discovering hidden factors of variation in deep networks. In *arXiv:1412.6583*.

Fischer, P.; Dosovitskiy, A.; Ilg, E.; Hausser, P.; Hazirbas, C.; and Golkov, V. 2015. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2758–2766.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*.

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

Hinton, G. E.; Krizhevsky, A.; and Wang, S. D. 2011. Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(0):1735–1780.

Hollingworth, A. 2004. Constructing visual representations of natural scenes: the roles of short- and long-term visual memory. *J. Exp. Psychol. Hum. Percept. Perform.* 30(3).

Horn, B., and Schunck, B. 1981. Determining optical flow. *Artificial Intelligence* 17:185–203.

Hurri, A., and Hyvarinen, A. 2003. Simple-cell-like receptive fields maximize temporal coherence in natural video. *Neural Computation* 15(3):663–691.

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

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; and Girshick, R. 2014. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Li, Y.; Paluri, M.; Rehg, J. M.; and Dollar, P. 2016. Unsupervised learning of edges. In *CVPR*.

Lin, W.; Zhou, Y.; Xu, H.; Yan, J.; Xu, M.; Wu, J.; and Liu, Z. 2017. A tube-and-droplet-based approach for representing and analyzing motion trajectories. *TPAMI*.

Liu, C.; Yuen, J.; and Torralba, A. 2011. Sift flow: Dense correspondence across scenes and its applications. *TPAMI*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

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*.

Mikolajczyk, K., and Schmid, C. 2005. A performance evaluation of local descriptors. *TPAMI* 27(10):1615–1630.

Mnih, V.; Heess, N.; Graves, A.; and Kavukcuoglu, K. 2014. Recurrent models of visual attention. In *NIPS*.

R. Goroshin, J. Bruna, J. T. D. E., and LeCun, Y. 2015. Unsupervised feature learning from temporal data. In *arXiv:1504.02518*.

Revaud, J.; Weinzaepfel, P.; Harchaoui, Z.; and Schmid, C. 2015. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*.

Revaud, J.; Weinzaepfel, P.; Harchaoui, Z.; and Schmid, C. 2016. Deepmatching: Hierarchical deformable dense matching. *IJCV*.

Srivastava, N.; Mansimov, E.; and Salakhutdinov, R. 2015. Unsupervised learning of video representations using lstms. In *ICML*.

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

van Hateren, J. H., and Ruderman, D. L. 1998. Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proceedings. Biological sciences / The Royal Society* 265(1412):2315–2320.

Wang, X., and Gupta, A. 2015. Unsupervised learning of visual representations using videos. In *ICCV*.

Weinzaepfel, P.; Revaud, J.; Harchaoui, Z.; and Schmid, C. 2013. Deepflow: Large displacement optical flow with deep matching. In *ICCV*.

Xu, L.; Jia, J.; and Matsushita, Y. 2012. Motion detail preserving optical flow estimation. *TPAMI* 34(9):1744–1757.

Zagoruyko, S., and Komodakis, N. 2015. Learning to compare image patches via convolutional neural networks. In *CVPR*.

Zbontar, J., and LeCun, Y. 2015. Computing the stereo matching cost with a convolutional neural network. In *CVPR*.

Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *ECCV*.

Zhou, T.; Krahenb, P.; Aubry, M.; Huang, Q.; and Efros, A. 2016. Learning dense correspondence via 3d-guided cycle consistency. In *CVPR*.