STFlow: Self-Taught Optical Flow Estimation Using Pseudo Labels

Zhe Ren, Wenhan Luo, Junchi Yan, *Member, IEEE*, Wenlong Liao, Xiaokang Yang, *Fellow, IEEE*, Alan Yuille, *Fellow, IEEE*, and Hongyuan Zha

***Abstract*— The Deep learning of optical flow has been an active area for its empirical success. For the difficulty of obtaining accurate dense correspondence labels, unsupervised learning of optical flow has drawn more and more attention, while the accuracy is still far from satisfaction. By holding the philosophy that better estimation models can be trained with better-approximated labels, which in turn can be obtained from better estimation models, we propose a self-taught learning framework to continually improve the accuracy using self-generated pseudo labels. The estimated optical flow is first filtered by bidirectional flow consistency validation and occlusion-aware dense labels are then generated by edge-aware interpolation from selected sparse matches. Moreover, by combining reconstruction loss with regression loss on the generated pseudo labels, the performance is further improved. The experimental results demonstrate that our models achieve state-of-the-art results among unsupervised methods on the public KITTI, MPI-Sintel and Flying Chairs datasets.**

***Index Terms*— Deep neural networks, optical flow, self-taught learning, unsupervised learning.**

I. Introduction

OPTICAL flow estimation has been a long-standing problem in computer vision. It can be simply understood as the motion field of pixels between two consecutive images. Since optical flow contains plenty of motion and geometry information, it has served as a building block for various applications, such as action recognition [1] and frame interpolation [2]. However, the problem has still not been fully explored for real-world applications.

Although traditional learning-free methods have made numerous achievements for a long time [3]–[5], being too time-consuming is a common issue that limits their practical applications, especially in time-critical scenarios. Due to the success of deep learning in different kinds of computer vision tasks, an increasing number of works concentrate on solving optical flow estimation by supervised deep learning [6]–[9]. These works employ a convolutional neural network (CNN) to infer the flow from image pairs in a single forward pass. Since the CNN models can be run efficiently on the GPU device, such approaches realize nearly real-time optical flow estimation. With the powerful computing hardware, deep learning based approaches (*e.g.* [7]) take a short time to be on par with the traditional methods in estimation accuracy, and even surpass [8], [9] their conventional counterparts on the most challenging benchmarks KITTI2015 [10] at present. However, supervised deep learning methods usually rely on a massive amount of labeled data. While for optical flow, accurate dense correspondence labels are expensive and difficult to acquire. As a result, existing methods turn to artificially synthesized datasets, which intrinsically exhibit unrealistic characteristics.

As there is no requirement of ground truth for training, researchers begin to explore unsupervised learning methods. Based on differentiable bilinear interpolation [12], the current unsupervised learning framework for optical flow estimation usually warps a reference image based on the estimated optical flow to reconstruct the target image [11], [13]. With local smoothness constraint, CNN for optical flow estimation could be trained successfully in a completely unsupervised way. However, the result is far from satisfaction. Occlusion is one of the major issues for unsupervised optical flow methods because the photometric constancy assumption is severely violated over occluded regions. For the sake of addressing occlusion, one popular strategy in practice is ignoring the reconstruction loss over occluded regions with an occlusion mask inferred by forward-backward consistency validation [14], forward warping [15] or CNN estimation [16]. As a result, smoothness constraint becomes the only supervision signal for unsupervised occlusion estimation. Although the assumption of uniform motion in [16] provides complementary information for flow estimation over occluded regions, motion with constant velocity assumption is still too strict for the

Manuscript received August 28, 2019; revised June 4, 2020 and August 1, 2020; accepted August 31, 2020. Date of publication September 21, 2020; date of current version September 25, 2020. This work was supported in part by NSFC under Grant 61972250, Grant U19B2035, and Grant U1609220; in part by the National Key Research and Development Program of China under Grant 2018AAA0100704; in part by STCSM under Grant 18DZ1112300; and in part by NSF under Grant 1763705. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Julian Fierrez. *(Corresponding author: Junchi Yan.)*

Zhe Ren and Xiaokang Yang are with the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: sunshinezhe@sjtu.edu.cn; xkyang@sjtu.edu.cn).

Wenhan Luo is with the Tencent AI Lab, Shenzhen 518000, China (e-mail: whluo.china@gmail.com).

Junchi Yan is the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yanesta13@163.com).

Wenlong Liao is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: igoliao@sjtu.edu.cn).

Alan Yuille is with the Department of Computer Science and Cognitive Science, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: alan.l.yuille@gmail.com).

Hongyuan Zha is with the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: zha@cc.gatech.edu).

Digital Object Identifier 10.1109/TIP.2020.3024015

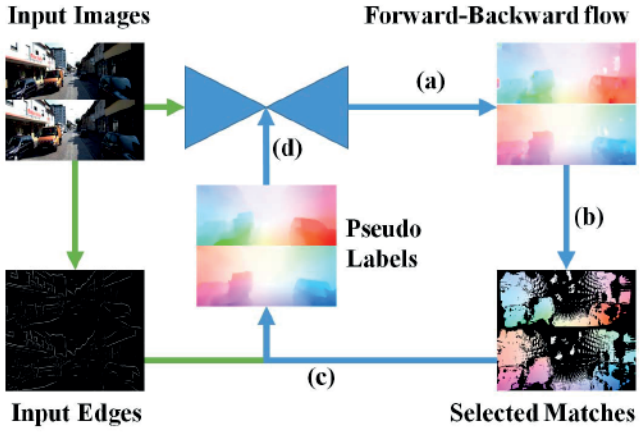


Fig. 1. Overview of our self-taught flow network learning framework using self-generated pseudo labels. One can adopt any flow net backbone (we adopt PWCNet in our experiments) for flow estimation via the unsupervised loss in [11]. The output forward-backward flows (a) from a flow network are checked with bidirectional consistency to generate the selected matches (b) which are further interpolated (using Epicflow with additional edge map inputs) to obtain dense flows (c). Such flows are expected to be better than the raw output flow and are used as pseudo labels for the self-taught learning iteratively (d).

real scene. Recently, the series methods [17], [18] propose to create data with the occlusion artificially, which is, in essence, a kind of complex data augmentation. Although it offers strong supervision over the occluded area, the occlusion pattern still can not be as natural as in the real scene.

Inspired by Epicflow [4], we introduce an edge-aware interpolation technique to reasonably infer the optical flow over occluded regions. In fact, previous studies [5], [19]–[22] have adopted Epicflow as a key component to estimate the final dense optical flow. What they actually do is to provide as accurate as possible sparse matches. Different from these methods, the sparse matches we used are actually from a trainable deep model for flow estimation. The interpolation is implemented to generate better-approximated labels, termed as `pseudo labels`, which are used to train the deep model in turn. This naturally forms an iterative procedure to train an optical flow estimation network in a `self-taught` manner.

Specifically, we propose a self-taught framework for deep learning based optical flow estimation to iteratively improve the performance without any ground truth. After initialing with an existing unsupervised method (we use [11] in this study), we select from the flow estimation by the deep model (Fig. 1(a)) a set of qualified matches (Fig. 1(b)) and further interpolate better dense optical flow (Fig. 1(c)) inspired by Epicflow. The interpolated optical flow with strong supervised signal over occluded regions is, in turn, used as pseudo ground truth to train the deep model for better optical flow estimation (Fig. 1(d)). Because of the learning ability of the deep model, the quality of selected matches improves gradually, which means more and more accurate pseudo labels are generated and higher and higher accuracy of the deep model for optical flow estimation could consequently be achieved. In this self-taught way, the accuracy of the deep model could be boosted as the iteration converges. In terms of the inference, our method still enjoys the speed advantage of deep learning models, which is applied with a single forward pass, without iteration.

This study pushes forward the frontier of unsupervised optical flow network learning by bridging the unsupervised learning models with a supervised learning paradigm, using the so-called pseudo labels generated by the to-be-taught model itself. Perhaps more importantly, we show how the mixed learning pipeline can be carefully designated and iteratively executed to boost the flow estimation performance.

In a nutshell, the main contributions of this paper are:

1) To the best of our knowledge, this is one of the first works showing how the self-taught framework can be adopted for unsupervised deep learning of optical flow estimation.

2) We improve the self-estimated flow by an edge-aware interpolation technique to provide pseudo labels for training a deep network. Combined with reconstruction loss, the self-taught process persistently boosts the deep network until satisfactory performance is achieved.

3) Our algorithm achieves state-of-the-art performance on public benchmarks, compared with peer unsupervised optical flow learning methods. In certain cases, it even outperforms supervised learning networks, *e.g.* FlowNetS [5].

The rest of the paper is organized as follows. Section II briefly reviews the literature on optical flow estimation. The details of the proposed approach are presented in Section III. Section IV reports the experimental results, and conclusive remarks are made in Section V.

II. Related Work

In this section, we briefly introduce existing approaches including learning-free methods and deep learning based methods for optical flow estimation.

A. Learning-Free Optical Flow Estimation

Traditional flow estimation methods are typically free of learning. The seminal work can be traced back to [23], which first introduces the photometric constancy. [24] assumes that motion in a neighborhood keeps constant so that optical flow can be computed locally. In [25], the authors propose a multi-scale warping model and prove such a scheme actually implements a coarse-to-fine warping strategy, within which the error is prone to propagate from coarse to fine scale. LDOF [26] combines discrete matching of the sparse descriptor with continuous optimization. Meanwhile, with the emergence of the PatchMatch technique [27] for fast approximate of nearest-neighbor matching, a series of works [5], [19], [28], [29] adopt it to address the large displacement problem. By imitating deep convolutional approaches, DeepMatching [30] is proposed to handle non-rigid deformations and repetitive textures. Deepflow [31] strengthens its ability to solve the large displacement problem by combining LDOF [26] with DeepMatching. Relying on sparse correspondences, Epicflow [4] uses edge information to calculate geodesic distance and interpolates the missing flow following geometrical structure. By using the consistency of forward and backward optical flow and symmetry of occlusion, Mirrorflow [3] estimates optical flow and occlusion map simultaneously. However, typical learning-free methods are usually time-consuming and cannot fully exploit the useful data.

B. Deep Learning Based Flow Estimation

1) Supervised Learning Methods: FlowNet [6] is the first work to model the optical flow estimation as a deep learning problem. A pair of images are fed into the network and dense optical flow is directly estimated as the output. SpyNet [32] estimates optical flow with a coarse-to-fine strategy. The prediction from the previous coarser level is the initialization of the subsequent finer level. In each level, only the residual flow is estimated. In order to further improve the performance, FlowNet2.0 [7] assembles multi FlowNet models with additional subnetwork specializing on small motions. For the first time, a deep learning based method achieves on par results with state-of-the-art learning-free methods. Simultaneously, two powerful and compact networks, PWC-Net [9] and LiteFlownet [8] are designed, both of which adopt the encoder-decoder structure. They construct a pyramid of features of two input images first and then compute the correlation between the feature of one image and warped feature of the other with the estimated flow from the previous level. Both networks perform on par with FlowNet2.0 [7], with fewer parameters. Recently [33] raises a more efficient and compact network by simplifying the volumetric layer with multi-channel cost volumes and separable volumetric filters. Besides directly estimating optical flow via deep neural networks, another strategy is to use the learnable deep features with classic patch matching approaches. FlowFieldCNN [20] adopts a thresholded loss to train a Siamese network to obtain more distinctive patch-based features and then apply the learned features to FlowField [5]. Although its performance is competitive, it still inherits the drawback of classic methods, *i.e.*, slow in inference.

2) Unsupervised Learning Methods: Because accurate dense correspondence labels are expensive to obtain, more and more attention has been recently shifted to unsupervised approaches. BackToBasic [13] and DSTFlow [11] are two pioneer works that attempt to train FlowNet end to end via classic reconstruction loss and smoothness regularization. Under the framework of DSTFlow, occlusion handling is incorporated in the Unflow [14], where an occlusion mask is reasoned by a consistency check between forward and backward flows. After fine-tuning on the KITTI training dataset, Unflow even achieves competitive accuracy against supervised methods on KITTI2012 and KITTI2015. OccAwareFlow [15] raises more accurate occlusion reasoning by modeling the non-occluded region as the range of backward optical flow [34]. DF-Net [35] jointly learns the depth and optical flow and makes two tasks benefit from each other by cross-task consistency. Instead of estimating optical flow from two images, MultiFrame-Flow [16] estimates optical flow by using three consecutive frames, assuming that occlusion is complementary between flow from the future frame and from the past frame. Recent state-of-the-art methods, DDFlow [18] and SelFlow [17] both adopt a self-supervised framework with a teacher model and a student model. They distill the flow estimation and use them as the labels for the hallucinated occlusion. Compared with the DDFlow that creates the occlusion simply by cropping, SelFlow randomly selects the superpixels as the occluded regions. Different from their teacher-student framework, our method derives an iterative self-taught learning framework.

One closely related work is [36], which guides flow network learning via third-party-generated labels. However, the presented technique is simplistic, and no iterative self-taught learning is performed, with neither quality control nor dense-flow interpolation. Moreover, as is shown in experiments, our method notably outperforms this Guidedflow baseline [36].

III. Self-Taught Flow Network Learning

A. Approach Overview

As shown in Fig. 1, we present a novel approach for deep learning based optical flow estimation by iteratively generating pseudo labels, in a self-taught manner. We first initialize an optical flow network on dataset $\mathcal{D}$ following the unsupervised framework of DSTFlow [11] and denote the trained flow network (or estimator)[1] as $M_{of}(\theta_0)$. After initialization, we begin to boost its ability iteratively. At the k-th round, we use flow estimator $M_{of}(\theta_{k-1})$ running on the whole dataset $\mathcal{D}$ to estimate both forward and backward dense optical flow F_k and B_k for each pair of images I_1 and I_2. See Fig. 1(a). Then, by a consistency check between F_k and B_k, two sets of qualified estimation $\tilde{F}_k$ and $\tilde{B}_k$ from F_k and B_k respectively are selected. See Fig. 1(b). Following Epicflow [4], an edge-aware sparse-to-dense interpolation is adopted to recover the missing flow, especially over occluded regions. After an additional one-scale refinement by the classical variational model [25], the updated dense flow estimation is obtained with improved accuracy. See Fig. 1(c). In particular, we treat these updated optical flows as approximate (pseudo) ground truth $\tilde{F}^{psd}$ and $\tilde{B}^{psd}$ for forward and backward flow, respectively, which are subsequently employed to train a better flow network $M_{of}(\theta_k)$. See Fig. 1(d). For a better illustration, a working example of generating pseudo labels is shown in Fig. 2.

Naturally, such a self-taught learning process can be repeated until the accuracy is converged. The algorithm is shown in Algorithm 1. In the following, we will detail each part.

B. Flow Estimator Backbone and Initialization

In general, the proposed self-taught framework is orthogonal to the choice of flow networks. In our practice, we primarily consider the popular network structure `PWCNet`, which is used by state-of-the-art unsupervised deep learning methods [16]–[18]. PWCNet is a compact and powerful network, which is composed of feature extractor, cost volume computing and flow estimator. It is a multi-scale convolutional neural network with an encoder-decoder structure. There are no other constraints on the resolution of input images except that the width and height of the input should be a multiple of 64. For more details, please refer to the work [9]. In our method, we employ it as the flow estimator, which is trained using our proposed self-taught flow learning algorithm.

[1] In this paper, we use the term *estimator* and *network* interchangeably.

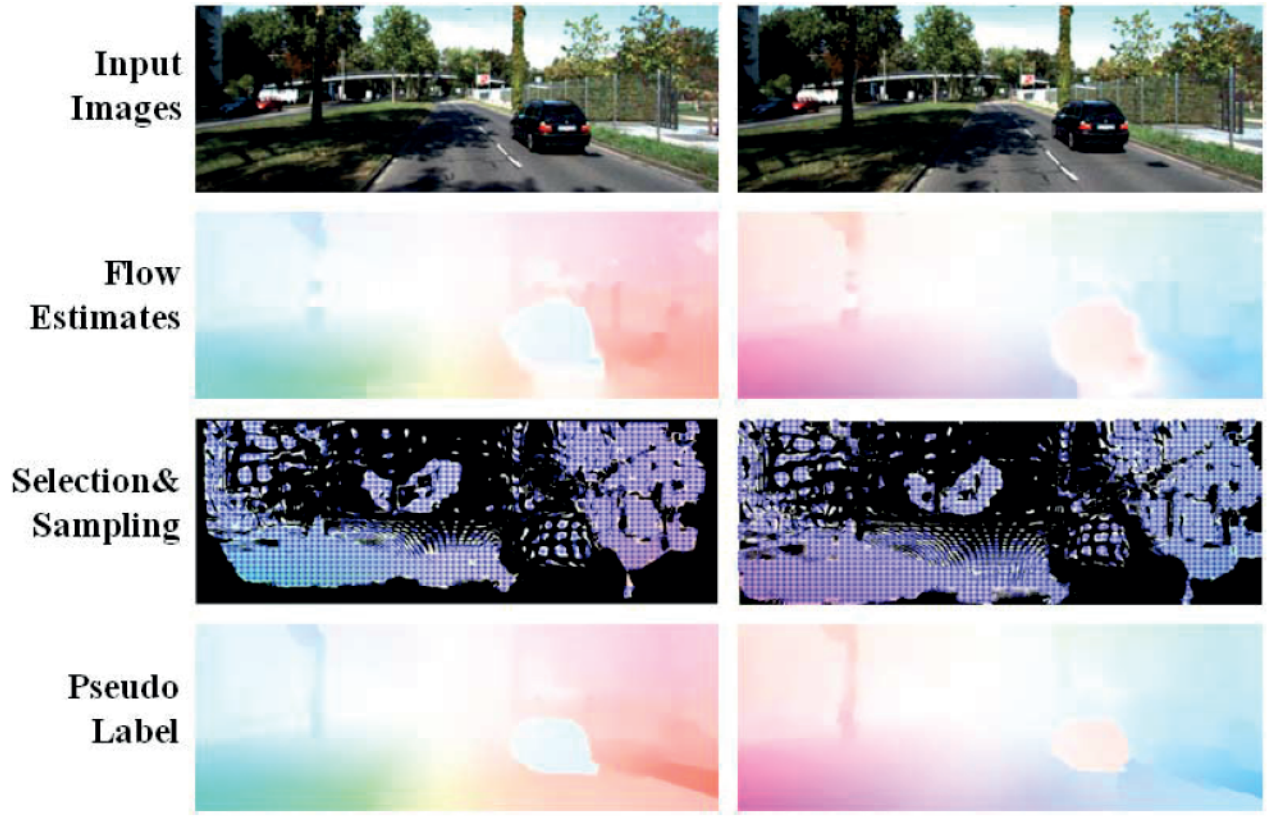


Fig. 2. An example of our pseudo label generation procedure. Starting with the estimated flow from the input images using an existing unsupervised learning model, *e.g.* [11]; Qualified matches are then selected by bidirectional consistency validation. After further locally sampling, sparse matches are selected to interpolate the enhanced flow, which is regarded as the pseudo labels used for the subsequent flow estimation.

Algorithm 1 Self-Taught Learning Flow Networks Using Pseudo Labels (**STFlow**)

Require: Image pairs (I_1, I_2) and the corresponding edge maps (e_1, e_2); unsupervised optical flow estimator $M_{of}(\theta)$ (DSTFlow [11] used in this study with the backbone: PWCNet [9])

Ensure: Optimized optical flow estimator parameters θ

1: Initialize $M_{of}(\theta)$ by running the unsupervised optical flow method DSTFlow [11]
$\theta_0 = DSTFlow(I_1, I_2, M_{of}(\theta))$
2: **for** k in $\{1, \cdots, K\}$ **do**
3: Estimate forward and backward flows on dataset $\mathcal{D}$ indexed with round k:
$F_k = M_{of}(\theta_{k-1}, I_1, I_2)$
$B_k = M_{of}(\theta_{k-1}, I_2, I_1)$
4: Select qualified matches conditioned by Eq. (4) and (5)
$\tilde{F}_k = Consistency\_check(F_k, B_k, I_1, I_2)$
$\tilde{B}_k = Consistency\_check(B_k, F_k, I_2, I_1)$
5: Generate dense flow from $\tilde{F}_k$, $\tilde{B}_k$ by Epicflow [4]
$\tilde{F}_k^{psd} = Epicflow(Sampling(\tilde{F}_k), e_1, I_1, I_2)$
$\tilde{B}_k^{psd} = Epicflow(Sampling(\tilde{B}_k), e_2, I_2, I_1)$
6: Train $M_{of}(\theta_k)$ by $\tilde{F}_k^{psd}, \tilde{B}_k^{psd}$
$\theta_k = Supervised\_training(M_{of}(\theta), \tilde{F}_k^{psd}, \tilde{B}_k^{psd})$
7: **end for**

We initialize the optical flow estimator $M_{of}(\theta)$ by adopting the unsupervised learning method DSTFlow [11] with minor modifications. Under the assumption of photometric constancy, image $\tilde{I}_2$ warped from I_2 based on forward optical flow F should be constant with image I_1. Typically, this is employed as the reconstruction loss for the unsupervised optical flow learning. The same differentiable warping technique in DSTFlow is also adopted to make sure of the gradient backpropagation. Because of the consistency check for matches selection in our approach, the reconstruction loss between I_2 and I_1 based on backward optical flow B is also considered. In order to reduce the computation cost, we do not use the gradient constancy in DSTFlow. Thus the data term is

$$\ell_{data}(I_1, I_2, F, B) = \sum_{\mathrm{x_i}} \underbrace{\Psi\left(I_2(\mathrm{x_i} + F(\mathrm{x_i})) - I_1(\mathrm{x_i})\right)}_{\text{appearance discrepancy by forward flow}} + \underbrace{\Psi\left(I_1(\mathrm{x_i} + B(\mathrm{x_i})) - I_2(\mathrm{x_i})\right)}_{\text{appearance discrepancy by backward flow}}, \quad (1)$$

where $F = M_{of}(\theta, I_1, I_2)$ and $B = M_{of}(\theta, I_2, I_1)$, and $\mathrm{x_i}$ denotes each point in image. Note that $\Psi(s) = (s^2 + C)^\gamma$, which is the Charbonnier penalty function [37] and we set $C = 1e^{-6}$ throughout the paper. Following the state-of-the-art methods [17], [18], we also adopt the Census Transform [38].

Besides, a smoothness regularization term is also utilized by minimizing the flow difference between each pixel and its neighbors. In addition, to reduce the effect over the motion boundary, each neighbor is weighted by the distance in the

Lab color space, as the following,

$$\ell_{smooth}(F,B)=\sum_{\mathrm{x_i}}\sum_{\mathrm{n}\in\mathcal{N}(\mathrm{x_i})}\omega(I_1^{Lab},\mathrm{x_i},\mathrm{n})\Psi\left(F(\mathrm{x_i})-F(\mathrm{n})\right)+\omega(I_2^{Lab},\mathrm{x_i},\mathrm{n})\Psi\left(B(\mathrm{x_i})-B(\mathrm{n})\right), \quad (2)$$

where $\mathcal{N}(\mathrm{x_i})$ is the collection of 4-connected neighbors of point $\mathrm{x_i}$ in vertical and horizontal directions. I_1^{Lab} is image I_1 represented in the *Lab* color space, which is perceptually uniform with the color variation. $\omega(I_1^{Lab},\mathrm{x_i},\mathrm{n})=\exp(\frac{-\|I_1^{Lab}(\mathrm{x_i})-I_1^{Lab}(\mathrm{n})\|^2}{\sigma^2})$. The weight ω is used to discourage flow smoothness in areas where the appearance changes dramatically, *e.g.*, edges.

The initialized parameter of the optical flow estimator is,

$$\theta_0=\arg\min_{\theta}\ell_{data}(\theta)+\alpha\cdot\ell_{smooth}(\theta), \quad (3)$$

where α is a weighting hyper-parameter.

C. Selection of Qualified Matches

Our principle is mainly based on the forward-backward consistency and photometric constancy. Ideally, for good matches, the forward optical flow should be the same as the backward flow in magnitude but with opposite direction. Besides, the intensity of the two corresponding pixels should also be as similar as possible. At each round, we select qualified matches from the estimation of the trained optical flow estimator $M_{of}(\theta)$. We estimate both forward and backward optical flow F and B for each pair of images (I_1, I_2) in the dataset $\mathcal{D}$. With forward optical flow F, the correspondence point (x_2, y_2) in I_2 for each point (x_1, y_1) in I_1 is $(x_2, y_2) = (x_1, y_1) + F(x_1, y_1)$. We then measure the forward-backward consistency for forward flow F by the constraint as follows,

$$\begin{cases}\dfrac{\|(x_2,y_2)+B(x_2,y_2)-(x_1,y_1)\|}{\|F(x_1,y_1)\|}<\epsilon_1, \\ \qquad\qquad if\quad \|F(x_1,y_1)\|\neq 0, \\ \dfrac{\|(x_2,y_2)+B(x_2,y_2)-(x_1,y_1)\|}{0.5}<\epsilon_1, \\ \qquad\qquad if\quad \|F(x_1,y_1)\|=0, \end{cases} \quad (4)$$

where $\|\cdot\|$ is the ℓ_2 norm. The photometric constancy is measured by the condition as,

$$\|I_1(x_1,y_1)-I_2(x_2,y_2)\|<\epsilon_2. \quad (5)$$

Note that since (x_2, y_2) may not be integer. Both $B(x_2, y_2)$ and $I_2(x_2, y_2)$ are obtained by bilinear interpolation.

For B, we follow the same principle. All the matches in F and B that meet the above conditions in Eq. (4) and (5) will be selected as the qualified flows $\tilde{F}$ and $\tilde{B}$, respectively.

D. Edge-Aware Interpolation

Based on selected matches discussed in Section III-C, we adopt the sparse-to-dense interpolation from Epicflow [4] to generate better flow estimation. Obviously, the matches we choose are all from the non-occluded region. If we consider only these matches, there will be no clue about occluded regions. Most existing unsupervised learning methods [14], [15], [35] handle the occlusion issue by using an estimated occlusion mask to remove the side effect of data term in Eq. (1) over the occluded region, where photometric constancy is violated. In this way, the only valid supervised signal for occlusion comes from the smoothness loss in Eq. (2), which requires that flow should be similar to its 4-connected neighbors. However, such a smoothness constraint for occlusion is not sufficiently informative and it also blurs the motion boundary. Recently, the works [17], [18] utilize the distilled optical flow as the labels for the occluded region. However, the occlusion of their data is created artificially, whose characteristic is different from that of the real scene. In this study, we resort to Epicflow to interpolate the missing flow by following the geometric structure of the scene of the image. Based on the edges of the image, an affine transform is estimated by weighting with a geodesic distance, which will be larger if more edges are traversed. As a result, the interpolated flow provides a strong supervised signal over the occluded region. Unlike the works [18], [17], our method directly addresses the occlusion in the real scene.

In practice, we do not interpolate directly from the selected matches $\tilde{F}$ and $\tilde{B}$ but further make them sparse by a local sampling. On one hand, too dense matches bring a heavy burden to interpolation. On the other hand, because the selected matches $\tilde{F}$ and $\tilde{B}$ still contain the outliers and estimation noise, a reasonable number of matches introduce less noise for pseudo label generation. We divide an image into several $h\times w$ non-overlapped blocks and in each block we select $\lfloor\tau\times h\times w+0.5\rfloor$ (round up) matches with least consistency error computed by Eq. (4). After sampling, the selected matches $\tilde{F}$ and $\tilde{B}$ are used as the anchor points to generate the dense optical flow by interpolation. Like Epicflow [4], we extract edges using the SED detector [39] and implement an extra step of one-scale variational energy optimization to further refine the interpolated optical flow. The final generated optical flow is treated as the approximate (pseudo) ground truth $\tilde{F}^{psd}$ and $\tilde{B}^{psd}$ (also termed as pseudo labels in this study), which are used to train a better optical flow estimator $M_{of}(\theta_k)$ in a self-taught manner.

E. Cooperation of Pseudo Supervised Loss With Unsupervised Reconstruction Loss

So far, the performance of the optical flow estimator can be boosted iteratively by repeating the generation process of the pseudo ground truth $\tilde{F}^{psd}$ and $\tilde{B}^{psd}$. Since the approximated ground truth is still noisy, we add the additional reconstruction loss in Eq. (1) to cooperate with our pseudo supervised loss. During training, two masks S_f and S_b with value $\{\mathbf{0},\mathbf{1}\}$ are computed by constraint Eq. (4), where $\mathbf{1}$ indicates the corresponding match satisfies the forward-backward consistency. In other words, the reconstruction loss is only imposed on the consistency region by weighting with non-occluded masks S_f and S_b. By doing so, not only the side effect of the outliers in the approximated ground truth is suppressed, but also the estimation over consistency regions is improved. In the end,

TABLE I

AVERAGE EPE (END POINT ERRORS) AND FL SCORE (*i.e.* OUTLIER RATIO – SEE MORE DETAILS IN THE CAPTION OF TABLE V) ON KITTI. 'BASELINE-C' POINTS TO THE BASELINE MODEL WITH CENSUS TRANSFORM. 'OUR' DENOTES 'STFLOW'. '-NO/R' MEANS NO RECONSTRUCTION LOSS. '−NO/V' INDICATES NO VARIATIONAL ENERGY OPTIMIZATION

Method	KITTI2015				KITTI2012		
	ALL	NOC	OCC	Fl-all	ALL	NOC	OCC
Baseline	13.67	6.58	41.28	33.18%	6.51	2.62	28.27
Baseline-C	8.21	3.32	26.96	20.18%	4.02	1.28	19.30
Our-no/r	4.06	2.34	11.24	14.56%	1.82	1.07	5.95
Our-no/v	3.69	2.15	10.23	11.88%	1.65	0.96	5.56
Our	3.56	2.08	9.95	11.58%	1.64	0.96	5.48

TABLE II

AVERAGE EPE (END POINT ERRORS) ON KITTI WITH DIFFERENT EDGE DETECTORS

Edge	KITTI2015				KITTI2012		
	ALL	NOC	OCC	Fl-all	ALL	NOC	OCC
SED [39]	3.56	2.08	9.95	11.58%	1.64	0.96	5.48
$\|\nabla_2 I\|$	3.85	2.21	10.85	12.14%	1.7	0.97	5.73

the final loss function $\ell_{train}(\theta)$ is as follows,

$$\ell_{train}(\theta) = \sum_{x_i} \underbrace{\|F(x_i) - \tilde{F}^{psd}(x_i)\| + \|B(x_i) - \tilde{B}^{psd}(x_i)\|}_{\text{pseudo supervised loss}} + \underbrace{\zeta \cdot S_f(x_i)\Psi\left(I_2(x_i + F(x_i)) - I_1(x_i)\right)}_{\text{forward reconstruction loss}} + \underbrace{\zeta \cdot S_b(x_i)\Psi\left(I_1(x_i + B(x_i)) - I_2(x_i)\right)}_{\text{backward reconstruction loss}}, \quad (6)$$

where $\|\cdot\|$ is ℓ_2 norm. $F = M_{of}(\theta, I_1, I_2)$ and $B = M_{of}(\theta, I_2, I_1)$. The smoothness loss term in Eq. (2) is not used since the dense pseudo ground truth provides smoothness implicitly.

IV. EXPERIMENTS

Our approach is named as STFlow for its self-taught nature. We benchmark our approach on three popular datasets, KITTI [40], [41], MPI-Sintel [42], and Flying Chairs [6], for optical flow estimation. On each dataset, we adopt PWCNet as the backbone and train a specific model for each dataset. 'EPE' and 'Fl' are used as the metrics for the optical flow evaluation. 'EPE' stands for the average end point errors, which is the average Euclidean distance over the pixels between the estimated flow and the ground truth. 'Fl' denotes the percentage of the outliers among the estimation. An estimate is counted as correct if the EPE of this estimate is < 3px or $< 5\%$ of its ground truth. In the paper, plenty of experiments have been done to prove the effectiveness of our method. In Section IV-D, a thorough ablation study is conducted to show the effect of each component in our approach. Affect of edge detector is investigated in Section IV-E and the analysis of convergence is shown in IV-F. Section IV-G compares the results of using pseudo labels from EpicFlow and pseudo labels from our method. In section IV-H, we further discuss the impact of the hyperparameters, ϵ_1 and ϵ_2. Extensive comparison with existing methods is conducted in Section IV-I.

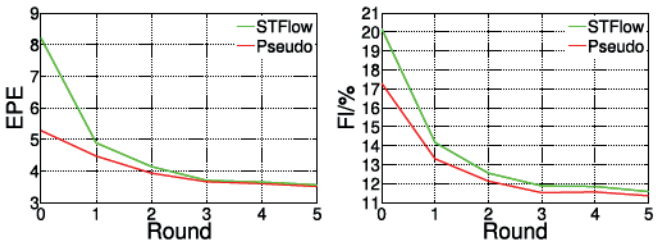


Fig. 3. EPE (in pixels) and Fl score (*i.e.* outlier ratio – see more details in the caption of Table V) of each round on KITTI2015. 'Pseudo' means pseudo labels generated and tested on the training set of KITTI2015. One can note the error in terms of both EPE and Fl decreases as the training rounds proceed, suggesting a convergence.

TABLE III

COMPARISON WITH THE RESULTS BY USING THE OUTPUT OF EPICFLOW AS THE PSEUDO LABELS IN OUR METHOD. 'EPIC-LABELS' DENOTES PSEUDO LABELS ARE FROM EPICFLOW METHOD. 'ST-LABELS' MEANS OUR METHOD WHERE PSEUDO LABELS COME FROM SELF-TAUGHT LEARNING

Method	KITTI2015			KITTI2012	
	TRAIN		TEST	TRAIN	TEST
	ALL	Fl-all	Fl-all	ALL	ALL
Epic-Labels	6.89	18.84%	-	2.67	-
ST-Labels	3.56	11.58%	13.83%	1.64	1.9

TABLE IV

AVERAGE EPE (END POINT ERRORS) ON FLYING CHAIRS WITH DIFFERENT ϵ VALUE

HyperParameter	ϵ_1			ϵ_2		
	0.4	0.5	0.6	10	20	30
EPE	2.56	2.53	2.56	2.53	2.53	2.54

A. Datasets

KITTI is a realistic car driving dataset, which consists of two benchmarks KITTI2012 [40] and KITTI2015 [10]. Compared with KITTI2012 including only the static scene, KITTI2015 is more challenging with dynamic scenarios. Following the previous works [11], [15], [17], we make the multi-view version data (without ground truth) as the training set and also exclude the images that exist in the KITTI benchmarks (with ground truth) and their two neighboring frames.

MPI-Sintel is generated from an open-source animated short film with three levels of rendering effects. Like other works [6], [11], [15], we first pre-train on the Flying Chairs and then finetune on the data from the *Clean* and *Final* training set together.

Flying Chairs is a synthetic benchmark by overlapping chairs rendered from 3D CAD models [43] on random background images from Flickr. We apply the same split in [6] for training and test.

B. Setting Details

For training, the Adam method [44] is used with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The batch size is set as 4. For the initialization

TABLE V

AVERAGE END POINT ERRORS *i.e.* EPE (IN PIXELS) OVER ALL, OCCLUDED (OCC) AND NON-OCCLUDED (NOC) AREAS OF DIFFERENT METHODS ON THE KITTI DATASET. THE METHOD STFLOW DENOTES TRAINING STFLOW USING 'KITTI' DATASET. HERE FL-ALL DENOTES THE 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. THE NUMBERS IN THE PARENTHESES ARE THE RESULTS OF THE NETWORKS ON THE DATA THEY WERE TRAINED ON, AND HENCE ARE NOT DIRECTLY COMPARABLE TO OTHER RESULTS AS THEY TEND TO OVERFIT. THE PERFORMANCE NUMBERS OF PEER METHODS ARE DIRECTLY QUOTED FROM THE ORIGINAL PAPERS AND THE DASH DENOTES UNREPORTED. THE BEST RESULTS AMONG UNSUPERVISED METHODS ARE IN BOLD AND OUR METHOD PERFORMS THE BEST IN EVERY CASE AMONG THE UNSUPERVISED DEEP LEARNING METHODS

Method \ Dataset	KITTI2015					KITTI2012			
	Train				Test	Train			Test
	ALL	NOC	OCC	Fl-ALL	Fl-ALL	ALL	NOC	OCC	ALL
Epicflow [4]	-	-	-	-	26.29%	3.47	-	-	3.8
Mirrorflow [3]	-	-	-	9.93%	10.29%	-	-	-	2.6
FlowNetS [5]	-	-	-	-	-	8.26	-	-	-
FlowNetS+ft [5]	-	-	-	-	-	7.52	-	-	9.1
FlowNet2 [7]	10.06	-	-	30.37%	-	4.09	-	-	-
FlowNet2+ft [7]	(2.3)	-	-	(8.61%)	10.41%	(1.28)	-	-	1.8
PWCNet [9]	10.35	-	-	33.67%	-	4.14	-	-	-
PWCNet+ft [9]	(2.16)	-	-	(9.8%)	9.6%	(1.45)	-	-	1.7
DSTFlow [11]	16.79	6.96	-	36%	39%	10.43	3.29	-	12.4
GuidedFlow [36]	-	-	-	-	-	-	-	-	9.5
Unflow-CSS [14]	8.1	-	-	23.27%	-	3.29	1.26	-	-
OccAwareFlow [15]	8.88	-	-	-	31.2%	3.55	-	-	4.2
MultiframeFlow [16]	6.59	3.22	19.11	-	22.94%	-	-	-	-
DF-Net [35]	8.98	-	-	26.01%	25.7%	3.54	-	-	4.4
DDFlow [18]	5.72	-	-	-	14.29%	2.35	-	-	3.0
SelFlow [17]	4.84	-	-	-	14.19%	1.69	-	-	2.2
STFlow	**3.56**	**2.08**	**9.95**	**11.58%**	**13.83%**	**1.64**	**0.96**	**5.48**	**1.9**

of optical flow estimators, we empirically set $\alpha = 2$, $\gamma = 0.45$, and $\sigma = 10$. We set $\epsilon_2 = 20$, $\tau = 0.05$ for qualified match selection, followed by sampling with $h = w = 4$. Each optical flow estimator is trained for five rounds ($K = 5$). In each round, we adopt the three-step learning schedule as in [6], which halves the learning rate at $\frac{1}{2}, \frac{2}{3}, \frac{5}{6}$ of the maximum iterations. The initial learning rate is set as 10^{-4} in the first four rounds and 10^{-5} in the last round. Data augmentation is also conducted following [7].

KITTI. We set $\epsilon_1 = 0.05$ for qualified match selection. Interpolation is implemented at the resolution of 384×1152. We train 180K iterations in each round with cropping resolution of 320×1152 from 384×1226. The testing is executed at the resolution of 384×1216. ζ is 1 for loss at the largest scale and decreases with the same proportion of the output resolution reduction at other scales.

Flying Chairs. $\epsilon_1 = 0.5$. In each round, we train 240K iterations and the cropping size for training is 320×448. Interpolation and testing are executed on the original size, 384×512. $\zeta = 0.01$ at the largest scale and decreases in the same way as in KITTI.

MPI-Sintel. $\epsilon_1 = 0.5$. We use the model trained on Flying Chairs as initialization and finetune with 60K iterations in each round. The cropping size is 384×960 and the interpolation resolution is 384×896. ζ is 0.1 at the largest scale and decreases like in KITTI.

C. Training Time

Usually, training time depends on many factors, like batch size, GPU performance, resolution of the input, network structure and the training iterations. In our experiments, we train our models on the Titan Xp GPU. Generally, it cost 32 hours per round for KITTI and 21 hours per round for Flying Chairs. For MPI-Sintel, 10.6 hours is spent in each round.

D. Ablation Study

In order to further show the effectiveness of each component in our methods, we conduct an ablation study on the KITTI dataset with PWCNet as a backbone network. The baseline is trained with losses in Eq. (1) and Eq. (2), which is the basic framework of unsupervised optical flow learning [11].

TABLE VI

AVERAGE END POINT ERRORS *i.e.* EPE (IN PIXELS) OVER ALL, OCCLUDED (OCC) AND NON-OCCLUDED (NOC) AREAS OF DIFFERENT METHODS ON THE FLYING CHAIRS AND MPI-SITNEL DATASETS. THE METHODS STFLOW DENOTES TRAINING STFLOW USING 'FLYING CHAIRS' AND 'MPI-SINTEL' DATASETS RESPECTIVELY. THE NUMBERS IN THE PARENTHESES ARE THE RESULTS OF THE NETWORKS ON THE DATA THEY WERE TRAINED ON, AND HENCE ARE NOT DIRECTLY COMPARABLE TO OTHER RESULTS AS THEY TEND TO OVERFIT. THE PERFORMANCE NUMBERS OF PEER METHODS ARE DIRECTLY QUOTED FROM THE ORIGINAL PAPERS AND THE DASH DENOTES UNREPORTED. THE BEST RESULTS AMONG UNSUPERVISED METHODS ARE IN BOLD AND OUR METHOD PERFORMS THE BEST IN ALMOST EVERY CASE. NOTE THAT THE RESULT OF THE SELFLOW INVOLVES USING THE TEST DATA IN THE SINTEL MOVIE

Method \ Dataset	Chairs	MPI-Sintel Clean				MPI-Sintel Final			
	Test	Train			Test	Train			Test
	ALL	ALL	NOC	OCC	ALL	ALL	NOC	OCC	ALL
Epicflow [4]	2.94	2.4	-	-	4.12	3.7	-	-	6.29
Mirrorflow [3]	-	-	-	-	3.32	-	-	-	6.07
FlowNetS [5]	2.71	4.5	-	-	7.42	5.45	-	-	8.43
FlowNetS+ft [5]	3.04	(3.66)	-	-	6.96	(4.44)	-	-	7.76
FlowNet2 [7]	-	2.02	-	-	3.96	3.14	-	-	6.02
FlowNet2+ft [7]	-	(1.45)	-	-	4.16	(2.01)	-	-	5.74
PWCNet [9]	-	2.55	-	-	-	3.93	-	-	-
PWCNet+ft [9]	-	(1.7)	-	-	3.86	(2.21)	-	-	5.13
DSTFlow [11]	5.11	6.93	5.05	-	10.4	7.82	5.97	-	11.11
GuidedFlow [36]	3.01	-	-	-	-	-	-	-	7.96
Unflow-CSS [14]	-	-	-	-	-	7.91	-	-	10.22
OccAwareFlow [15]	3.3	(4.03)	-	-	7.95	(5.95)	-	-	9.15
MultiframeFlow [16]	-	(3.89)	(2.64)	(11.21)	7.23	(5.52)	(4.32)	(12.87)	8.81
DDFlow [18]	2.97	(2.92)	-	-	6.18	3.98	-	-	7.4
SelFlow [17]	-	**(2.88)**	-	-	6.56	(3.87)	-	-	**6.57**
STFlow	**2.53**	**(2.91)**	**(1.9)**	**(8.71)**	**6.12**	**(3.59)**	**(2.57)**	**(9.6)**	**6.63**

Census Transform is also implemented, which has been proved effective in the [14], [17], [18]. We further develop two variants of our method. One is our full method without the reconstruction loss in Eq. (6), and the other is our full method without the one-scale variational energy optimization step during the pseudo label generation process. All the counterparts are trained under the same configuration.

Results in Table I suggest that, besides the effectiveness of the census transform, all variants of our methods improve the results by a large margin, especially on the 'OCC' region. In general, both the reconstruction loss and the variational energy optimization step contribute to improving the estimation performance. From the third row and the fifth row, the cooperation of two losses significantly benefits the self-taught learning of flow estimator on all the scenarios. The comparison between the fourth row and the fifth row shows that the additional variational energy optimization step also slightly improves the results in all cases.

E. Edge Dectector

Additionally, we also investigate how the performance is affected when using different types of edge detectors. Specifically, we compare the employed SED detector with the norm of the gradient of the images $\|\nabla_2 I\|$. Two experiments are conducted on KITTI datasets under the same setting but using different edge detectors for the pseudo label generation.

Table II reports the comparison results. Even though using the simple edge detector, image gradient, our method still obtains reasonable results on both benchmarks. It reveals that the performance of our method is not sensitive to the choice of edge detector.

F. Convergence Analysis

To clarify the boosting process of our method, we present the intermediate results of each round on the KITTI2015 training set. Besides the accuracy of estimation, we also show the accuracy of generated pseudo labels as the upper-bound performance at each round. Note that, because there is no ground truth for the multi-view version data that we actually use for training, we have to test the pseudo labels on the original training set. The results are shown in Fig. 3, regarding both the EPE in pixels and the Fl score. As the training rounds proceed, the pseudo labels with better accuracy are generated and our method gradually improves the network performance in both the EPE and the Fl metrics. To make progress more evident, we also provide the visualization of the estimated optical flow at different rounds in Fig. 4 and Fig. 5. Similarly, as the self-taught learning goes, better and

Fig. 4. Qualitative results of the estimated flow fields on KITTI2015 dataset. Original input images are shown at the top row. From the second to the fifth row, the estimation of the initialization, the first round, third round and the fifth round are displayed respectively. Ground truth is shown at the bottom row. The visualization is performed by using the flow visualization protocol in [42].

better results are estimated (like the cars in KITTI and the background in MPI-Sintel).

G. Comparison With EpicFlow

In order to further show the benefits of our self-taught learning framework, we conduct experiments with different label resources. One is directly using the output of the EpicFlow [4] as the pseudo labels for training. The other is using our generated pseudo labels by our self-taught learning. All of the rest settings are kept the same.

As shown in Table III, our method is significantly superior to the counterpart method on all the metrics. The main reason is that Epicflow acquires the correspondence by DeepMatching [30] method, whose matching ability is limited and fixed. On the contrary, our method obtains the correspondence by the estimation of the self-taught neural network. As the training proceeds, correspondence used to generate pseudo labels in our method becomes more and more accurate.

H. Impact of ϵ

ϵ_1 and ϵ_2 in Eq. (4) and Eq. (5) determine the quality of the selected matches, which will influence the accuracy of the generated pseudo label. In order to show the impact of each hyperparameter in our method, we implement the experiments on the Flying Chairs dataset with various ϵ values. The default setting of ϵ_1 is 0.5 and ϵ_2 is 20.

As is shown in Table IV, the result with the default setting is the best among the various cases. When varying both ϵ values near the default number, our method still obtains the reasonable results with nearly no decline. It proves that our method is robust on the choice of the ϵ value and all the values within a nearby range of the default number are acceptable. Compared with ϵ_2, ϵ_1 influences more on the final result. We believe the optimal ϵ value should be adaptively decided by the different training samples and training phases, and we leave this problem in the future study.

I. Comparison With Existing Methods

In this part, we compare our method with several milestone works in the optical flow field, including state-of-the-art unsupervised learning methods, such as SelFlow [17], DDFlow [18], MultiframeFlow [16], supervised learning methods like PWCNet [9] and classic learning-free methods like Mirrorflow [3].

In Table V and Table VI, we conduct extensive experiments on the Flying Chairs, KITTI and MPI-Sintel datasets. Note that, Epicflow and Mirrorflow are traditional learning-free methods. FlowNetS(2/+ft), PWCNet(+ft) are supervised learning methods. DSTFlow, GuidedFlow, Unflow, OccAwareFlow, MultiframeFlow, DF-Net, DDFlow, SelFlow and the proposed STFlow are the unsupervised methods.

The results suggest that,

Fig. 5. Qualitative results of the estimated flow fields on MPI-Sintel dataset. Original input images are shown at the top row. From the second to the fifth row, the estimation of the initialization, the first round, third round and the fifth round are displayed respectively. Ground truth is shown at the bottom row. The visualization is performed by using the flow visualization protocol in [42].

1) In general, STFlow achieves state-of-the-art results among the unsupervised deep learning methods on all three benchmarks. Note that the result of SelFlow on the MPI-Sintel involves using the Test data in the Sintel movie while we only adopt the training set.

2) On KITTI, the EPE result of STFlow on the KITTI2012 test set surpasses that of Mirrirflow, which is the state-of-the-art learning-free estimator in KITTI2012. It is worth noting that our results are even competitive with those of supervised methods like FlowNet2.

3) On MPI-Sintel, STFlow also beats the supervised FlowNetS and FlowNetS+ft. However, the results of our methods are still not on par with those of Epicflow.

4) On Flying Chairs, our model STFlow is superior to all the unsupervised deep learning methods and even surpasses the supervised method FlowNetS.

J. Qualitative Results

Qualitative results of the STFlow model on KITTI2015 and MPI-Sintel datasets are shown in Fig. 4 and Fig. 5. The visualization is performed by using the flow visualization protocol in [42], where intensity represents the magnitude of the flow and the hue of colors represents the direction. By comparing the flow of the fifth round (fifth row) with the initial flow estimation (second row), our method improves the quality of the flow estimation significantly. As the iteration goes, more detail becomes clear. In KITTI2015 dataset, the motion edge of the car becomes sharper and the occluded part and the background become accurate increasingly. In MPI-Sintel, the motions over the occluded region (*e.g.*, motion of wing in the first column and the motion near the edge of the broadsword in the third column) are still be recovered successfully.

V. Conclusion

We propose an iterative self-taught learning framework for optical flow estimation. Our method requires no labor-extensive ground truth labels but tries to perform supervised learning via self-generated pseudo labels. We develop an effective pseudo label generation and learning pipeline which involves, i) initial flow field generation by pre-trained unsupervised flow network, ii) persistently improved pseudo label generation via bidirectional quality check and edge-aware interpolation; iii) supervised flow network learning using pseudo labels and cooperated with reconstruction loss. These steps are iteratively conducted in a self-taught learning manner. As the iteration goes, the performance of the network improves increasingly with better pseudo labels generated. A comprehensive ablation study has been done and the evaluation of public benchmarks verify the effectiveness and robustness of the proposed approach.

References

[1] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[2] S. Meyer, A. Djelouah, B. McWilliams, A. Sorkine-Hornung, M. Gross, and C. Schroers, "PhaseNet for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 498–507.

[3] J. Hur and S. Roth, "MirrorFlow: Exploiting symmetries in joint optical flow and occlusion estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 312–321.

[4] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "EpicFlow: Edge-preserving interpolation of correspondences for optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1164–1172.

[5] C. Bailer, B. Taetz, and D. Stricker, "Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4015–4023.

[6] A. Dosovitskiy *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[7] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2017, p. 6.

[8] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A lightweight convolutional neural network for optical flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8981–8989.

[9] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.

[10] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3061–3070.

[11] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, "Unsupervised deep learning for optical flow estimation," in *Proc. AAAI*, vol. 3, 2017, p. 7.

[12] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[13] J. Y. Jason, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 3–10.

[14] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss," 2017, *arXiv:1711.07837*. [Online]. Available: http://arxiv.org/abs/1711.07837

[15] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu, "Occlusion aware unsupervised learning of optical flow," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4884–4893.

[16] J. Janai, F. Güney, A. Ranjan, M. Black, and A. Geiger, "Unsupervised learning of multi-frame optical flow with occlusions," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 690–706.

[17] P. Liu, M. Lyu, I. King, and J. Xu, "SelFlow: Self-supervised learning of optical flow," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4571–4580.

[18] P. Liu, I. King, M. R. Lyu, and J. Xu, "DDFlow: Learning optical flow with unlabeled data distillation," in *Proc. AAAI*, 2019, pp. 8770–8777.

[19] Y. Hu, R. Song, and Y. Li, "Efficient coarse-to-fine patch match for large displacement optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5704–5712.

[20] C. Bailer, K. Varanasi, and D. Stricker, "CNN-based patch matching for optical flow with thresholded hinge embedding loss," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, vol. 2, no. 3, p. 7.

[21] Q. Chen and V. Koltun, "Full flow: Optical flow estimation by global optimization over regular grids," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4706–4714.

[22] M. Menze, C. Heipke, and A. Geiger, "Discrete optimization for optical flow," in *Proc. German Conf. Pattern Recognit.* Cham, Switzerland: Springer, 2015, pp. 16–28.

[23] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, nos. 1–3, pp. 185–203, Aug. 1981.

[24] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. IJCAI*, 1981, pp. 674–679.

[25] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2004, pp. 25–36.

[26] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.

[27] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[28] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1744–1757, Sep. 2012.

[29] L. Bao, Q. Yang, and H. Jin, "Fast edge-preserving PatchMatch for large displacement optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3534–3541.

[30] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "DeepMatching: Hierarchical deformable dense matching," *Int. J. Comput. Vis.*, vol. 120, no. 3, pp. 300–323, Dec. 2016.

[31] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1385–1392.

[32] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2017, p. 2.

[33] G. Yang and D. Ramanan, "Volumetric correspondence networks for optical flow," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 794–805.

[34] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez, "Symmetrical dense optical flow estimation with occlusions detection," *Int. J. Comput. Vis.*, vol. 75, no. 3, pp. 371–385, Sep. 2007.

[35] Y. Zou, Z. Luo, and J.-B. Huang, "DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 38–55.

[36] Y. Zhu, Z. L. Lan, S. Newsam, and A. G. Hauptmann, "Guided optical flow learning," in *Proc. CVPR Workshop Brave New Ideas Motion Spatio-Temporal Represent.*, 2017, pp. 1–5.

[37] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *Int. J. Comput. Vis.*, vol. 106, no. 2, pp. 115–137, Jan. 2014.

[38] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 1994, pp. 151–158.

[39] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1841–1848.

[40] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[41] M. Menze, C. Heipke, and A. Geiger, "Joint 3D estimation of vehicles and scene flow," in *Proc. ISPRS Workshop Image Sequence Anal. (ISA)*, 2015, pp. 1–8.

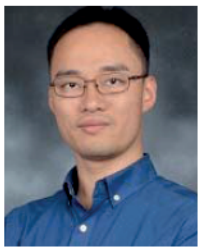
[42] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 611–625.

[43] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic, "Seeing 3D chairs: Exemplar part-based 2D-3D alignment using a large dataset of CAD models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3762–3769.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

Zhe Ren received the B.E. degree in electronics and information engineering from Northwestern Polytechnical University, Xi'an, China, in 2014. He is currently pursuing the Ph.D. degree with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. He also takes a Joint Ph.D. Program with the Johns Hopkins University, Baltimore, MD, USA. His current research interests include optical flow, deep learning, and unsupervised learning.

Wenhan Luo received the B.E. degree from the Huazhong University of Science and Technology, China, in 2009, the M.E. degree from the Institute of Automation, Chinese Academy of Sciences, China, in 2012, and the Ph.D. degree from the Imperial College London, U.K., 2016. He is currently working as a Senior Researcher with the Tencent AI Lab, China. His research interests include computer vision and machine learning, such as motion analysis (especially object tracking), image/video quality restoration, object detection, and recognition, reinforcement learning.



Junchi Yan (Member, IEEE) is currently a Research Professor (Ph.D. Advisor) with the Department of Computer Science and Engineering and the AI Institute of Shanghai Jiao Tong University. He is also the Co-Director for the prestigious SJTU ACM Class (in charge of AI direction). Before that, he was a Senior Research Staff Member with IBM Research–China, where he started his career since April 2011, and once an Adjunct Professor with the School of Data Science, Fudan University. His research interests include machine learning and computer vision. He serves as an Associate Editor for IEEE ACCESS, a (Managing) Guest Editor for the IEEE TRANSACTIONS ON NEURAL NETWORK AND LEARNING SYSTEMS, *Pattern Recognition Letters*, and *Pattern Recognition*, a Vice Secretary of China CSIG-BVD Technical Committee, and on the executive board of *ACM China Multimedia Chapter*. He has published more than 40 peer-reviewed articles in top venues in AI and has filed more than 20 U.S. patents. He has once been with the IBM Watson Research Center, Japan NII, and the Tencent/JD AI Lab as a Visiting Researcher. He was a recipient of the Distinguished Young Scientist of Scientific Chinese and the CCF Outstanding Doctoral Thesis.

Wenlong Liao received the B.E. degree in the major of detection, guidance, and control techniques from Northwestern Polytechnical University, in 2011, and the M.S. degree in control science and engineering from Shanghai Jiao Tong University in 2014, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering. Since then, he has been working on autonomous driving for specific scenarios. His research interests include robotics and computer vision.

Xiaokang Yang (Fellow, IEEE) received the B.S. degree from Xiamen University in 1994, the M.S. degree from the Chinese Academy of Sciences in 1997, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2000. He is currently a Distinguished Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. His research interests include visual signal processing and communications, media analysis and retrieval, and pattern recognition. He serves as an Associate Editor for the IEEE TRANSACTIONS ON MULTIMEDIA and an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS.

Alan Yuille (Fellow, IEEE) received the B.A. degree in mathematics from the University of Cambridge in 1976. His Ph.D. on theoretical physics, supervised by Prof. S.W. Hawking, was approved in 1981. He was a Research Scientist with Artificial Intelligence Laboratory, MIT and the Division of Applied Sciences, Harvard University, from 1982 to 1988. He served as an Assistant and Associate Professor at Harvard until 1996. He was a Senior Research Scientist with Smith-Kettlewell Eye Research Institute from 1996 to 2002. He was a Full Professor of Statistics with the University of California, Los Angeles, as a Full Professor with joint appointments in computer science, psychiatry, and psychology. He moved to Johns Hopkins University in January 2016. His research interests include computational models of vision, mathematical models of cognition, medical image analysis, and artificial intelligence and neural networks.

Hongyuan Zha received the Ph.D. degree in scientific computing from Stanford University in 1993. He is currently a Professor with the School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology, and the Guest Chair Professor with Shanghai Jiao Tong University. Since then, he has been working on information retrieval, machine learning applications and numerical methods. He was a recipient of the Leslie Fox Prize (1991, second prize) of the Institute of Mathematics and its Applications, the Outstanding Paper Awards of the 26th International Conference on Advances in Neural Information Processing Systems (NIPS 2013), and the Best Student Paper Award (advisor) of the 34th ACM SIGIR International Conference on Information Retrieval (SIGIR 2011). He was an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING and *Pattern Recognition*.