Deep Optical Flow Supervised Learning With Prior Assumptions

Article *in* IEEE Access · August 2018

DOI: 10.1109/access.2018.2863233

5 authors, including:

Xiang Xuezhi
Harbin Engineering University
93 PUBLICATIONS **793** CITATIONS
SEE PROFILE

Abdulmotaleb El Saddik
University of Ottawa
883 PUBLICATIONS **16,934** CITATIONS
SEE PROFILE

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 29, 2018, accepted August 1, 2018, date of publication August 6, 2018, date of current version August 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2863233

Deep Optical Flow Supervised Learning With Prior Assumptions

XUEZHI XIANG[1], (Member, IEEE), MINGLIANG ZHAI[1], RONGFANG ZHANG[1], YULONG QIAO[1], (Member, IEEE), AND ABDULMOTALEB EL SADDIK[2], (Fellow, IEEE)

[1]School of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China
[2]School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada

Corresponding author: Xuezhi Xiang (xiangxuezhi@hrbeu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61401113, in part by the Natural Science Foundation of Heilongjiang Province of China under Grant LC201426, and in part by the Fundamental Research Funds for the Central Universities of China under Grant HEUCF180801.

ABSTRACT Traditional methods for estimating optical flow use variational model that includes data term and smoothness term, which can build a constraint relationship between two adjacent images and optical flow. However, most of them are too slow to be used in real-time applications. Recently, convolutional neural networks have been used in optical flow area successfully. Many current learning methods use large data sets that contain ground truth for network training, which can make use of prior knowledge to estimate optical flow directly. However, these methods overemphasize the factor of deep learning and ignore advantages of many traditional assumptions used in variational framework for optical flow estimation. In this paper, inspired by classical energy-based optical flow methods, we propose a novel approach for dense motion estimation, which combines traditional prior assumptions with supervised learning network. During training, the variation in image brightness, gradient and spatial smoothness are embedded in network. Our method is tested on both synthetic and real scenes. The experimental results show that employing the prior assumptions during training can obtain more detailed and smoothed flow fields and can improve the accuracy of optical flow estimation.

INDEX TERMS Optical flow estimation, convolutional neural networks, supervised learning, prior assumptions.

I. INTRODUCTION

Optical flow estimation is one of the most fundamental problems in computer vision and has many applications, including autonomous driving, video segmentation and video semantic understanding. Classical approaches for estimating optical flow have achieved rapid progress in the last two decades. Most methods adopt the energy minimization approach proposed by Horn and Schunck [1]. These classical methods are based on a variational formulation and a related energy minimization problem. The classical energy function contains a data term that constrains the relationship between image pair and flow and contains a smoothness term that reflects the spatial information of the flow field in image. Based on the basic energy function, many improvements have been introduced in [2]–[4]. One of the challenges of optical flow estimation is to deal with the problem of large displacement. To address large displacement problem, coarse-to-fine scheme is employed in [7], which estimates initial optical flow at coarse level and warps the second image to the first image by using the up-sampled flow at fine level. Brox and Malik [8] add a matching term in variational framework for feature matching, which can provide point correspondences between images and moving objects. Inspired by the large displacement optical flow (LDOF) of Brox and Malik, DeepFlow [9] combines a matching algorithm with variational model for optical flow. The matching algorithm named DeepMatching [10] builds upon a multi-stage architecture which contains convolution and max-pooling operations. However, [9], [10] do not have any learnable parameters. The successive work EpicFlow [11] improves DeepFlow [9] by employing an interpolation method based on edge map, and by using energy minimization which contains a data term and a smoothness term as post-processing. Motion discontinuity preserving is also an issue in optical flow estimation. Nagel and Enkelmann [12] first uses anisotropic (image driven) smoothness assumption for raising the accuracy of optical flow estimation

at edge of motion. Sun *et al.* [13] analyses the principles in classical energy function and adds a non-local term to prevent over-smoothing across boundaries. Hua *et al.* [14] presents edge-aware constraints (EAC) which is added in a variational model for edge preserving. Tu *et al.* [15] improves [14] by employing an additional image fidelity term (IFT) to estimate optical flow and to restore image within a variational formulation. Traditional methods based on variational framework can build an energy model for optical flow estimation by using prior assumptions. However, these methods are usually computationally expensive. Most of them are too slow to be used in real-time applications.

Nowadays, convolutional neural networks (CNNs) have advanced a variety of computer vision tasks due to their immense learning capacity and superior efficiency. The CNNs have the ability to approximate the complex, non-linear transformation between the input and output. Example applications include activity recognition [16], [17], classification [18], semantic segmentation [19], object detection [20], and remoting sensing image processing [21]. Inspired by the successful application of deep learning in computer vision tasks, Dosovitskiy *et al.* [5] first proposes a CNN architecture named FlowNet to directly learn the optical flow from data end-to-end. At the same time, they publish a large synthetic dataset named Flying Chairs for training. The architecture of [5] is U-net which consists of a contracting part and an expanding part. Given adjacent frames as input, the network can predict dense optical flow directly. Moreover, as training loss, they use endpoint error (EPE), which is a standard error measure for optical flow estimation. However, EPE loss function is a simple constraint for optical flow estimation, which only calculates the error between ground truth and predictive flow and does not contain the relationship between image pairs and flow fields. The EPE loss also ignores traditional assumptions used in variational framework. In contrast to [5], in our framework, we take advantage of the traditional assumptions and define a novel loss function that contains three terms: supervised term, data term and smoothness term.

In this paper, inspired by traditional variational formulation for optical flow estimation, we consider the constraint relationship between image pairs and optical flow during training, and combine traditional assumptions with a supervised learning network. In our method, as shown in Fig. 1, brightness and gradient constancy assumptions are used to constrain the relationship between image pair and flow. And smoothness assumption is added in our loss layer to constrain the spatial information of flow. Compared with traditional optical flow framework, the superiority of learning method is that it can take advantage of prior knowledge and allows the network to learn an implicit prior from a large training dataset. In addition, we also use the endpoint error (EPE) to supervise our network like [5].

Once trained, given a pair of images as input, our CNN model gives an estimation of the motion field at its output layer. We train our CNN model on Flying Chairs dataset and test it on MPI-Sintel, KITTI2012, KITTI2015, Middlebury and Flying Chairs datasets. The experimental results on several public datasets show that our method can obtain more detailed and smoothed flow fields and outperforms [5] on several benchmarks.

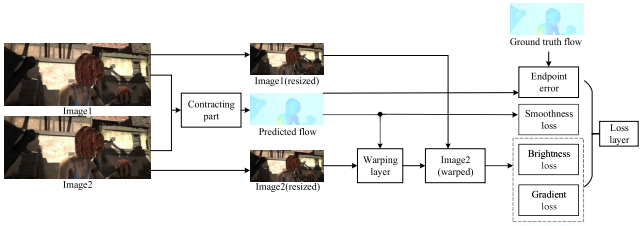


FIGURE 1. The proposed method. The presented network architecture consists of three key components: contracting part based on the structure of FlowNetS [5], warping layer based on spatial transform network [6] and loss layer based on supervised term and prior assumptions.

The rest of this paper is organized as follows. Section II reviews the recent methods on optical flow estimation. Section III-A focuses on the proposed network architecture. Section III-B mainly discusses the proposed loss function combining prior assumptions. In section IV, we present the experimental results. Finally, we give a brief conclusion in section V.

II. RELATED WORK

In this section, we provide a brief survey of methods based on traditional machine learning for optical flow estimation. And then, we mainly focus on methods based on deep learning in optical flow area.

A. MACHINE LEARNING-BASED OPTICAL FLOW METHOD

Before deep learning optical flow methods are proposed, several authors have applied machine learning techniques to optical flow estimation. Sun *et al.* [22] studies a statistical model using training sequences which contain image pairs and ground truth flow fields. However, the learning process is limited by the lack of training samples. Li and Huttenlocher [23] presents a method for learning optical flow by modelling a continuous-state Markov random field (MRF). Rosenbaum *et al.* [24] learns local statistics of optical flow using Gaussian mixture models.

B. DEEP LEARNING-BASED OPTICAL FLOW METHOD

Recently, convolutional neural networks (CNNs) have been used in optical flow estimation. Dosovitskiy *et al.* [5] first proposes two networks, FlowNetS and FlowNetC, for learning optical flow end-to-end, which take two consecutive input images and output a dense optical flow map using an encoder-decoder architecture. In this work, a large synthetic dataset for training named Flying Chairs is published, which is generated by rotation and translation. Based on [5], Vaquero [25] joints coarse-and-fine reasoning for deep optical flow learning by casting the task to a joint classification and regression problem. Mayer *et al.* [26] applies the architecture of [5]

in disparity and scene flow estimation, and publishes a large synthetic dataset for training disparity and optical flow. Teney and Hebert [27] presents a CNN architecture to learn dense optical flow from video sequence, which is trained on only 8 sequences of the Middlebury dataset. Ranjan and Black [28] combines spatial pyramid formulation with deep learning architecture for optical flow estimation, which estimates large motions on coarse layer and warps the second image toward the first one using the up-sampled flow from a previous level. Ilg *et al.* [29] improves the accuracy of optical flow estimation by stacking several sub-networks. This large network is first trained on FlyingChairs, and then is fine-tuned on FlyingThings3D proposed in [26]. However, the running time is increased due to the stacked operation. Zhang *et al.* [30] proposes a novel deep motion representation for understanding human behaviors in videos, which suppresses background noise in images and emphasizes human motion. Hu *et al.* [31] proposes a recurrent spatial pyramid network to deal with large displacement problem. [5], [25], [28], [31] all use Flying Chairs dataset as basic training set for supervised learning.

However, the supervised learning methods always require a large number of synthetic datasets for training. Obtaining dense flow ground truth for real scenes is usually difficult. To address this problem, other researchers present unsupervised learning methods for optical flow estimation. Ahmadi and Patras [32] presents a method training a CNN using the UCF101 dataset for motion estimation in an unsupervised manner, and using brightness constancy assumption with a Taylor series expansion to guide the training process. Yu *et al.* [33] designs an unsupervised network based on FlowNetS [5] architecture, which consists a loss function that combines brightness constancy assumption with a spatial term. Long *et al.* [34] trains a CNN for optical flow estimation by interpolating frames. Zhu and Newsam [35] extends DenseNet architecture [36] to a fully convolutional network (FCN) to learn motion estimation in an unsupervised manner. Ren *et al.* [37] presents an unsupervised learning network based on FlowNet architecture [5], which uses nonlinear data term and spatial smoothness to constrain the training process of the network. Although these unsupervised methods [32]–[35], [37] do not need to use large datasets as training samples, the accuracy is slightly inferior to the supervised approaches.

III. APPROACH

Given an image pair I_1, I_2 and ground truth F', our object is to train a CNN model that can estimate the per-pixel motion field F. The F' contains u and v which are horizontal and vertical displacements respectively. We first introduce our network architecture for optical flow estimation in Section III-A. Then, we provide details on how the extra assumptions are integrated with a CNN model in Section III-B.

A. NETWORK ARCHITECTURE

Many current networks for learning optical flow [25], [29], [32], [33], [35], [37] follow the basic architecture proposed in [5]. FlowNet created two networks, FlowNetS and FlowNetC. The FlowNetS contains two parts: contractive part and expanding part. In contractive part, two images are

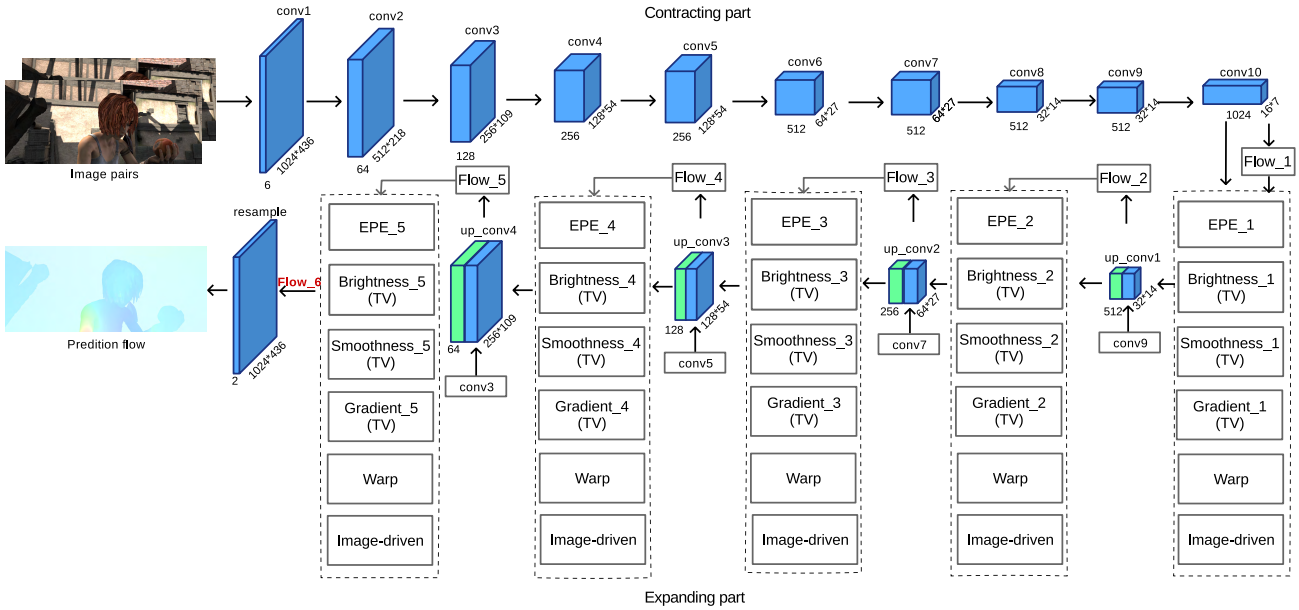


FIGURE 2. An overview of our proposed framework based on FlowNetS. The network architecture is based on FlowNetS, which is composed with a series of convolution layer and deconvolution layer. The contractive part is similar as FlowNetS. In the expanding part, after each deconvolution operation, the predicted optical flow value at the corresponding resolution is output, which is used to calculate the multi-assumption loss.

first stacked together and are fed through a series of convolution layers. Then, the last convolution layer generates two feature maps corresponding to the horizontal and vertical predicted optical flow with low resolution. Because we need to get dense flow field which has same resolution with the input two frames finally, successive deconvolutions are used to expand the flow in expanding part. Thus, per-pixel motion field can be estimated using CNN end-to-end. Another proposed network FlowNetC is to create two separate processing streams for the two images and to combine them using correlation layer which can find matching between the two streams.

In our framework, we employ multi-scale scheme to guide the supervised learning by down-sampling images to different smaller scales. we adopt FlowNetS and FlowNetC architectures and propose a multi-assumption supervised learning loss function which is added in the expanding part during the network training. Fig. 2 shows our network architecture based on FlowNetS. The contractive part contains successive convolutional layers with different strides (1 and 2). Fig. 3 shows the contractive part of FlowNetC. The expanding parts of FlowNetS and FlowNetC are same. In the expanding part, in contrast to FlowNetS and FlowNetC [5], our model not only employs the endpoint error (EPE) loss at different scales in to guide the learning of the network but also incorporates the prior assumptions which contain brightness constancy, gradient constancy and smoothness term to constrain the training of the network. These prior assumptions can be seen as extra auxiliary terms to guide the network training. The smoothness term is incorporated with an edge-aware formulation (image-driven), which takes advantage of the gradient of image to avoid over-smoothing at the edge of motion. All these prior assumptions are based on total variation (TV) regularization. The warping operation is employed to transform the second image to the first image using the predicted flow at different scales. To enable backpropagation of the warping layer, we adopt the spatial transformer module [6]. In particular, the warped image can be expressed as the following formula,

$$I_w = \sum_m^H \sum_n^W I_{mn} \max(0, 1 - |x_2 - n|) \max(1 - |y_2 - m|), \tag{1}$$

where the I_{mn} is the input image and the (m, n) are the coordinates in I_{mn}. (x_2, y_2) are the sampling coordinates in the second image. Backpropagation processing is defined by computing partial derivatives,

$$\frac{\partial I_w}{\partial I_{mn}} = \sum_m^H \sum_n^W \max(0, 1 - |x_2 - n|) \max(0, 1 - |y_2 - m|), \tag{2}$$

$$\frac{\partial I_w}{\partial x_2} = \sum_m^H \sum_n^W I_{mn} \max(0, 1 - |y_2 - m|) \times \begin{cases} 0, & if\ |n - x_2| \geq 1 \\ 1, & if\ n \geq x_2 \\ -1, & if\ n \leq x_2 \end{cases} \tag{3}$$

where $\frac{\partial I_w}{\partial y_2}$ can be computed in a similar way.

In summary, our network can be trained end-to-end. During training, we use data augmentation to avoid over-fitting. Given two consecutive images and corresponding ground truth, the network automatically calculates the losses caused in the training process and carries on the backpropagation. All the layer weights are learned end-to-end through backpropagation. During test, given an image pair, our network can output the predicted dense optical flow directly.

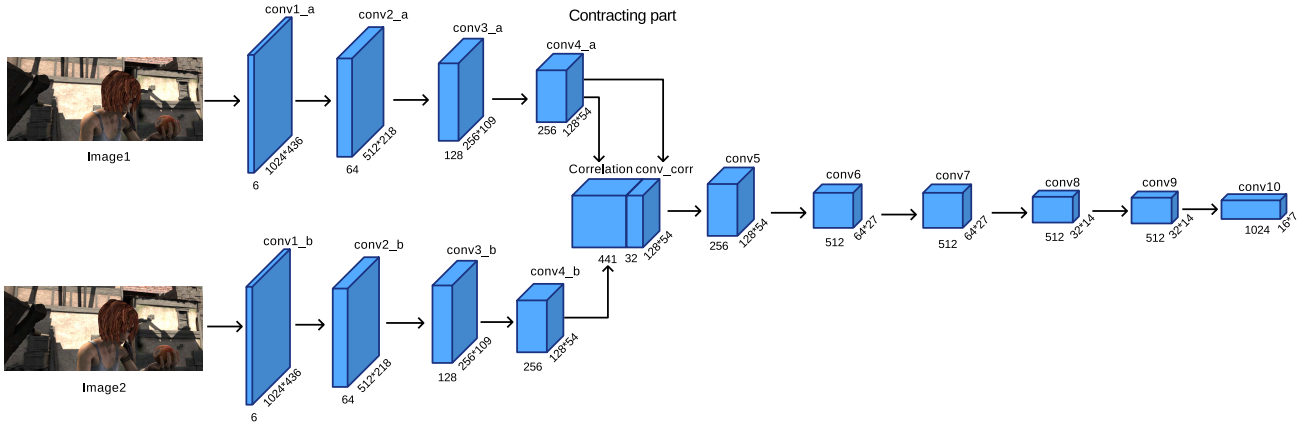


FIGURE 3. An overview of the contractive part of FlowNetC. The contractive part of FlowNetC has two streams that are fused by a correlation layer. The correlation layer can find matching between the two streams.

B. MULTI-ASSUMPTION SUPERVISED LEARNING

1) DATA TERM

The basic constraint of optical flow estimation is brightness constancy which assumes that the value of a pixel is not changed by the displacement,

$$I(x, y, t) = I(x + u, y + v, t + 1), \tag{4}$$

where $I(x, y, t)$ denotes the pixel value at point $x = (x, y)$ at frame t, and $w = (u, v, 1)^{\mathrm{T}}$ is the searched displacement vector between the frame t and another frame $t + 1$, u and v are horizontal and vertical displacements respectively.

In our framework, we put the brightness constancy assumption into our optimization process and define a loss function as Eq(5) to measure the predicted flow error during training,

$$L_b = \sum_{\mathbf{x}}^{N} \rho_D(\mid I_2(\mathbf{x} + w(\mathbf{x})) - I_1(\mathbf{x}) \mid), \tag{5}$$

where the N denotes the total number of pixels, and the ρ_D is robust penalty function. Here we use the Charbonnier penalty function $(x^2 + 0.001^2)^{\alpha}$.

The brightness constancy assumption has an obvious drawback which is quite susceptible to the slight changes in brightness. To address this issue, gradient constancy assumption is employed in many traditional methods for optical flow estimation, which can be given as following,

$$\nabla I(x, y, t) = \nabla I(x + u, y + v, t + 1), \tag{6}$$

where $\nabla = (\partial_x, \partial_y)^T$ denotes the spatial gradient.

In our framework, gradient constancy assumption is put into the network, which is defined as Eq(7),

$$L_g = \sum_{x}^{N} \rho_D(|\nabla I_2(x + w(x)) - \nabla I_1(x)|). \tag{7}$$

2) SMOOTHNESS TERM

Here, the model estimates the optical flow without taking any interaction between neighbor pixels into account, which neglects the spatial information in the image. Hence, it is useful to introduce the smoothness of the flow field as an extra assumption. Early smoothness assumption for optical flow is global smoothing as Eq(8) to assume a smooth flow field,

$$E_{smooth}(u, v) = min\{(\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2 + (\frac{\partial v}{\partial x})^2 + (\frac{\partial v}{\partial y})^2\}, \tag{8}$$

where $\frac{\partial u}{\partial x}$, $\frac{\partial u}{\partial y}$, $\frac{\partial v}{\partial x}$ and $\frac{\partial v}{\partial y}$ are respectively the gradients of the estimated optical flow (u, v) in the horizontal and vertical directions.

For the smoothness loss, we adopt an image-driven formulation, which can avoid over-smoothing at the edge of motion.

As shown in Eq(9), we define a smoothness loss that contains robust penalty function for decreasing outliers,

$$L_s = \rho_s\left(e^{-\alpha|\nabla I_1|}\frac{\partial u}{\partial x}\right) + \rho_s\left(e^{-\alpha|\nabla I_1|}\frac{\partial u}{\partial y}\right) + \rho_s\left(e^{-\alpha|\nabla I_1|}\frac{\partial v}{\partial x}\right) + \rho_s\left(e^{-\alpha|\nabla I_1|}\frac{\partial v}{\partial y}\right), \tag{9}$$

instead of using L_2 norm, we use Charbonnier penalty function ρ_S.

The derivative formulations of Eq(9) are shown as Eq(10) and Eq(11),

$$\frac{\partial L_s}{\partial u} = \sum_{i=1}^{W}\sum_{j=1}^{H} \frac{e^{-\alpha|\nabla I_1|}[(u_{i,j} - u_{i+1,j}) + (u_{i,j} - u_{i,j+1})]}{\{[(u_{i,j} - u_{i+1,j}) + (u_{i,j} - u_{i,j+1})]^2 + \varepsilon^2\}^{\alpha}} \tag{10}$$

$$\frac{\partial L_s}{\partial v} = \sum_{i=1}^{W}\sum_{j=1}^{H} \frac{e^{-\alpha|\nabla I_1|}[(v_{i,j} - v_{i+1,j}) + (v_{i,j} - v_{i,j+1})]}{\{[(v_{i,j} - v_{i+1,j}) + (v_{i,j} - v_{i,j+1})]^2 + \varepsilon^2\}^{\alpha}}, \tag{11}$$

where the W, H are width and height of image, the $u_{i,j}$, $v_{i,j}$ are horizontal and vertical flow in point (i, j).

3) SUPERVISED TERM

To supervise the network with ground truth flow, we employ the endpoint error (EPE), which is the standard error measure for optical flow estimation. The L_{epe} is shown as following,

$$L_{epe} = \sum_{i=1}^{W}\sum_{j=1}^{H} \sqrt{(u_{i,j} - u'_{i,j})^2 + (v_{i,j} - v'_{i,j})^2}, \tag{12}$$

where $u_{i,j}$, $v_{i,j}$ are the predicted flow fields at point (i, j) from the CNN and $u'_{i,j}$, $v'_{i,j}$ are ground truth flow fields at (i, j) respectively.

The total loss is a simple weighted sum of the brightness constancy loss, the gradient constancy loss, the smoothness loss, and the EPE loss,

$$L_{final} = \lambda_1 L_b + \lambda_2 L_g + \lambda_3 L_s + \lambda_4 L_{epe}, \tag{13}$$

where λ_1, λ_2, λ_3 and λ_4 weight the relative importance of loss terms during training. As shown in Fig. 2, we calculate the L_{final} at different resolution in the expanding part.

IV. EXPERIMENTAL RESULTS

In this section, we evaluate our methods on several optical flow benchmark datasets including MPI-Sintel, Flying Chairs, Middlebury, KITTI2012, and KITTI2015, and compare our results to existing traditional methods (non-learning) and learning methods (both supervised and unsupervised methods).

A. DATASETS FOR TRAINING AND EVALUATION

An overview of the used datasets is given in Table 1. Chairs and Things3D are the synthetically generated datasets. KITTI is a real-world dataset and has two versions: 2012 and 2015. MPI-Sintel is a synthetic dataset which contains two versions: clean and final. The following is a detailed introduction to these datasets.

1) FLYING CHAIRS

Flying Chairs is a synthetic dataset created by applying affine transformations to images collected from Flickr for optical flow learning. The dataset contains 22232 training image pairs and 640 test image pairs with ground truth flow.

TABLE 1. Overview of datasets.

Dataset	Chairs	MPI -Sintel	KITTI 2012	KITTI 2015	Flying Things 3D
Image pairs	22872	1593	389	400	26066
Training pairs	22232	1041	194	200	4248
Test pairs	640	552	195	200	21818
Synthetic	√	√	×	×	√
Dense ground truth	√	√	×	×	√

2) MPI-SINTEL

MPI-Sintel is also a synthetic dataset based on an animated movie and contains many large motions up to 400 pixels per. There are 1628 frames, 1064 for training and 564 for testing. It has two parts: clean and final. Clean contains realistic illuminations and reflections. Final additionally adds rendering effects like motion, defocus blurs and atmospheric effects.

3) KITTI OPTICAL FLOW 2012

KITTI optical flow 2012 is a real-world dataset created from a platform on a driving car and contains images of city streets. It consists of 194 training image pairs and 195 test pairs with sparse ground truth flow.

4) KITTI OPTICAL FLOW 2015

KITTI optical flow 2015 is a real-world dataset created from a platform on a driving car and contains images of city streets. It consists of 200 training scenes and 200 test scenes.

5) FLYINGTHINGS3D

FlyingThings3D is a large dataset to train convolution neural networks for disparity, optical flow and scene flow estimation. It consists of 25000 stereo frames with ground truth data and contains 2247 different scenes. This dataset is built to facilitate the training of large convolution neural networks.

B. TRAINING DETAIL

Our network was trained end-to-end using Adam optimization, where its parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We first use Flying Chairs datasets to train our network with 600k iterations. As the same with [5], we split the dataset into 22232 samples (i.e. image pairs) for training and 640 samples for testing, respectively. And, we started with learning rate $\lambda = 1e-4$ and divided it by 2 every 100k iterations after the first 300k. The pre-trained model is named "Ours-600k". We further trained model with long iterative process. We first used FlyingChairs dataset to train the network with 1200k iterations and set batch = 8. We started with learning rate $\lambda_c = 1e-4$ and divided it by 2 every 200k iterations after the first 400k. Then, we fine-tuned this pre-trained model on FlyingThings3D with 500k iterations. We started with learning rate $\lambda_t = 1e-5$ and divided it by 2 every 100k iterations after the first 200k. The model fine-tuned on FlyingThings3D is named "Ours-1700k". Due to the difference between the datasets for training and the datasets for testing, the adaptability of the model on some new test datasets will be reduced. To address this issue, we fine-tuned "Ours-600k" on Sintel training datasets (clean and final) for 8000 iterations using learning rate $\lambda_f = 1e-6$. The fine-tuned model is named "Ours-ft-600k". We added our prior

TABLE 2. Performance comparison on public benchmarks.

Methods	Sintel clean AEE		Sintel final AEE		KITTI2012 AEE		KITTI2015 Fl-all	Middlebury AEE	Flying Chairs AEE
	train	test	train	test	train	test	test	train	test
PCA-Flow [38]	-	6.83	-	8.65	-	**6.2**	-	-	-
HS [1]	-	8.74	-	9.61	-	9.0	69.60%	-	-
Classic+NL [13]	-	7.96	-	9.15	-	-	-	-	-
EPPM [39]	-	6.49	-	8.38	-	9.2	-	-	-
LDOF [8]	4.19	7.56	6.28	9.12	13.73	12.4	39.33%	0.45	3.47
UnsupFlowNet [33]	-	-	-	-	11.3	9.9	35.07%	-	5.30
USCNN [32]	-	-	-	8.88	-	-	-	-	-
FlowNet2.0-S [29]	3.79	-	4.93	-	-	-	-	-	-
FlowNet2.0-C [29]	3.04	-	4.29	-	-	-	-	-	-
DenseNetflow [35]	-	-	-	10.07	-	11.6	-	-	4.73
CaF-Full-41c [25]	6.51	9.42	7.28	10.18	-	-	-	-	3.18
SPynet [28]	4.12	6.69	5.57	8.43	9.12	-	**31.17%**	**0.33**	2.63
CNN-flow [27]	-	-	9.36	10.04	-	-	-	0.45	-
FlowNetS [5]	4.50	7.42	5.45	8.43	8.26	-	51.00%	1.09	2.71
FlowNetC [5]	4.31	7.28	5.87	8.81	9.35	-	-	1.15	2.19
FlowNetS+ft (Sintel) [5]	3.66	6.96	4.44	7.76	-	-	-	0.98	3.04
FlowNetC+ft (Sintel) [5]	3.78	6.85	5.28	8.51	-	-	-	0.93	2.27
DSTFlow [37]	6.93	10.40	7.82	11.11	16.98	-	52.00%	-	5.11
RecSpyNet [31]	-	-	6.69	9.38	10.02	13.7	40.90%	-	2.63
Ours-S-600k	3.92	7.28	5.02	8.19	7.90	9.0	49.74%	1.08	2.43
Ours-S-ft-600k (Sintel)	3.39	6.80	**4.09**	7.62	7.79	8.9	48.97%	1.13	2.95
Ours-C-600k	3.36	7.01	5.13	8.53	8.76	9.5	51.32%	1.10	2.05
Ours-S-1700k	3.47	6.59	4.85	7.78	6.12	7.6	47.95%	1.04	2.29
Ours-C-1700k	**2.91**	**6.44**	4.15	**7.47**	**5.77**	6.8	45.11%	0.99	**2.01**

TABLE 3. Parameter settings.

i^{th}	Scale	λ_1^i	λ_2^i	λ_3^i	λ_4^i	α
0	2^{-6}	0.1	0.1	1.0	0.32	0.4
1	2^{-5}	0.3	0.3	1.0	0.08	0.4
2	2^{-4}	0.5	0.5	1.0	0.02	0.45
3	2^{-3}	0.7	0.7	1.0	0.01	0.45
4	2^{-2}	1.0	1.0	1.0	0.005	0.45

TABLE 4. Ablation study.

Endpoint error	Brightness constancy	Gradient constancy	Smoothness assumption	Sintel clean train	Sintel final train
√				4.50	5.45
√	√			4.37	5.23
√	√	√		4.33	5.21
√	√		√	3.95	5.06
√	√	√	√	**3.92**	**5.02**

assumptions into FlowNetS and FlowNetC architectures. The model based on FlowNetS is named "Ours-S". The model based on FlowNetC is named "Ours-C". The experiments are performed on an Intel Xeon E5-1650 CPU, and an NVIDIA 1080 Ti GPU with batch size of 8. The hyper-parameters $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \alpha)$ were set at different stages of the expanding part as shown in Table 3.

C. RESULTS AND DISCUSSION

In this section, we compare our proposed method to recent state-of-the-art approaches on different benchmarks. The evaluation results on the training and testing sets are reported in Table 2. On MPI-Sintel, Middlebury, KITTI2012 and Flying Chairs, we use average endpoint error (AEE) as our criterion for error evaluation. On KITTI2015, we use "Fl-all" which is ratio of pixels where flow estimate is wrong by both ≥ 3 pixels and $\geq 5\%$. References [1], [8], [13], [38], [39] are not learning methods. References [5], [25], [27]–[29], [31]–[33], [35], [37] are learning methods based on CNNs. References [5], [25], [27]–[29], [31] are supervised learning methods. References [32], [33], [35], [37] are unsupervised learning approaches. Visual results are presented in Fig. 4, Fig. 6, Fig. 7 and Fig. 8 for MPI-Sintel dataset, KITTI2012 dataset and KITTI2015 dataset.

1) MPI-SINTEL

We evaluate the performance of our model on MPI-Sintel in two ways. First, we used Flying Chairs dataset to train our model directly, fine-tuned the pre-trained model on FlyingThings3D (optional) and evaluated our performance on Sintel clean and final datasets (both training and test sets). From the Table 2, we can find that our model "Ours-S-600k" outperforms FlowNetS [5] on Sintel training and test datasets (clean and final versions), and outperforms FlowNetC [5] on Sintel training set (clean version) and Sintel test set (clean and final versions). The model "Ours-C-600k" outperforms FlowNetC [5] on Sintel training set (clean version) and Sintel test set (clean and final versions). The results of "Ours-S-600k" and "Ours-C-600k" show the effectiveness of using prior assumptions. And our model "Ours-S-600k" outperforms another supervised methods [25], [27], [28], [31] on Sintel final dataset and Sintel clean (training set) dataset. For Sintel clean (test set), "Ours-S-600k" performs inferior to [28] (AEE 6.69). Compared with unsupervised methods [32], [35], [37], "Ours-S-600k" outperforms [32], [35], [37] on Sintel test set (final version). "Ours-S-1700k" outperforms most learning methods such as [25], [28], and [31]–[33] on Sintel training and test dataset. "Ours-C-1700k" outperforms learning methods [5], [25], [27]–[29], [31], [32], [35], [37] on Sintel training and test set. And from the Table 2, we can find that the "Ours-C-1700k" can achieve the best results on Sintel clean and final (test set). In Fig. 8, we compared "Ours-C-1700k" with "FlowNet2.0-C" and visualized the results. Second, we fine-tuned our model on Sintel (clean and final versions). The fine-tuned model is listed as "+ft" in Table 2. We compared our fine-tuned model with "FlowNetS+ft" and "FlowNetC+ft" [5]. The results show that "Ours+ft" outper-

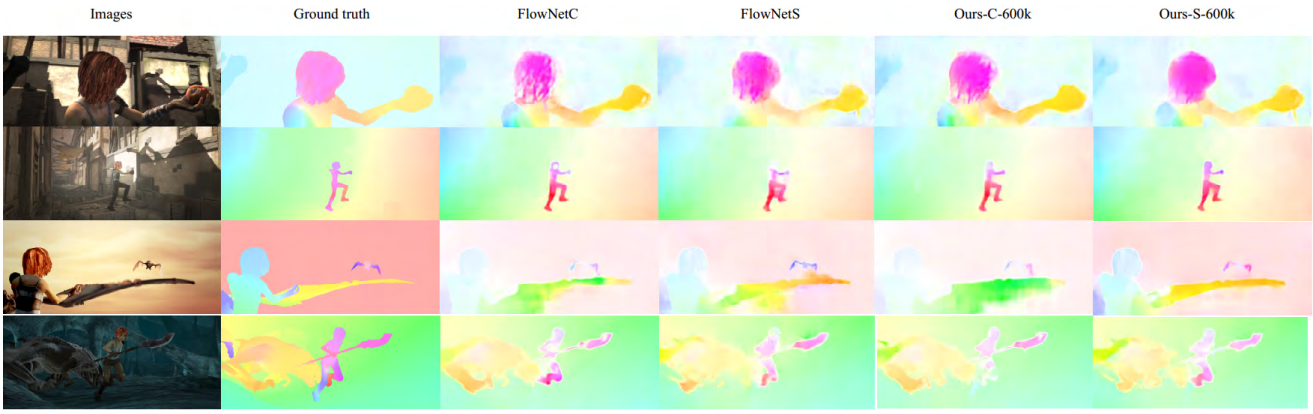


FIGURE 4. Examples of optical flow estimation from different methods using MPI-Sintel dataset (final version). In each row left to right: image, ground truth flow and 4 predictions: FlowNetC, FlowNetS, "Ours-C-600k" model and "Ours-S-600k" model. Note that our method can obtain more smoothed and detailed flow.

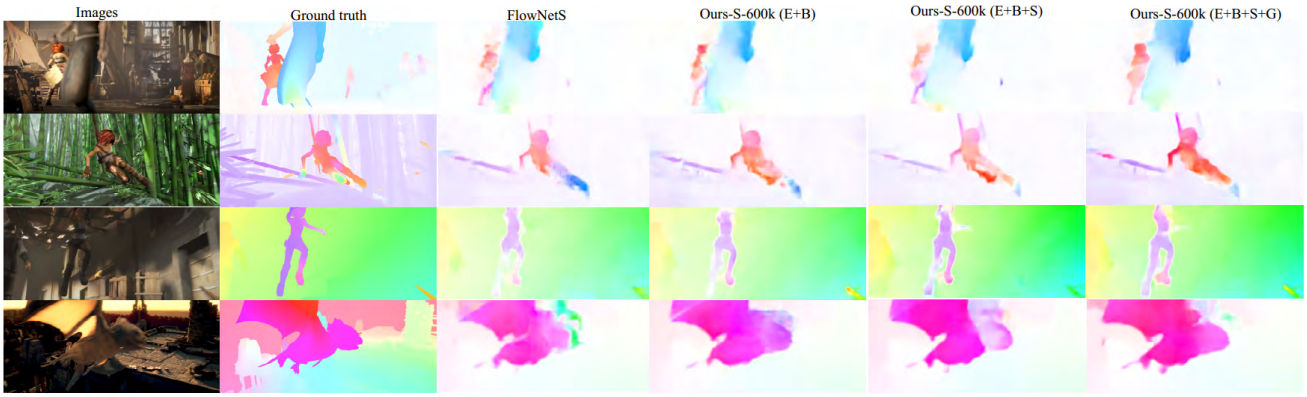
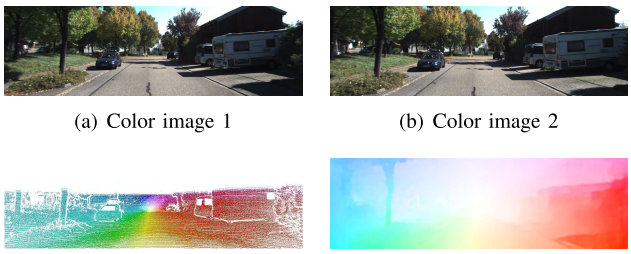


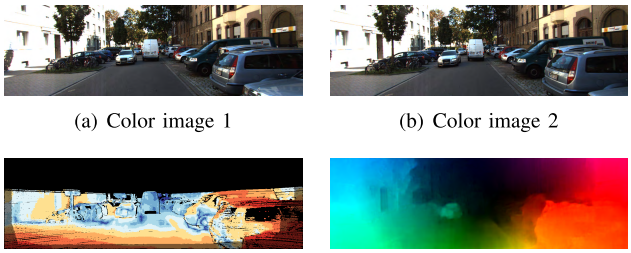
FIGURE 5. Examples of optical flow estimation from different prior assumptions using MPI-Sintel dataset (final version). The "E+B" denotes using endpoint error and brightness constancy assumption. The "E+B+S" denotes using endpoint error, brightness constancy assumption and smoothness assumption. The "E+B+S+G" denotes using endpoint error, brightness and gradient constancy assumption and smoothness assumption. In each row left to right: image, ground truth flow and 4 predictions: FlowNetS, Ours-S-600k (E+B), Ours-S-600k (E+B+S) and Ours-S-600k (E+B+S+G). Note that all our methods improve the results.



(a) Color image 1 (b) Color image 2

(c) Ground truth flow map (d) Flow map (Ours-S-600k)

FIGURE 6. Example of optical flow estimation on KITTI2012 training dataset. (a) is first color image. (b) is second color image. (c) is the sparse ground truth flow. (d) is flow map.



(a) Color image 1 (b) Color image 2

(c) Error map(our) (d) Flow map (Ours-S-600k)

FIGURE 7. Example of optical flow estimation on KITTI2015 test dataset. (a) is first color image. (b) is second color image. (c) is the error map which depicts correct estimates ($\geq$ 3 px or $\geq$ 5% error) in blue and wrong estimates in red tones. (d) is flow map.

forms "FlowNetS+ft" and "FlowNetC+ft". Visual results on the Sintel final dataset are presented in Fig. 4. To visualize the flow fields, we used the tool provided by Sintel [40]. In Fig. 4, we compared our model "Ours-S-600k" and "Ours-C-600k" with FlowNetS and FlowNetC. Our results are more detailed and smoothed due to the brightness, gradient and smoothness (image-driven) assumptions.

2) KITTI AND MIDDLEBURY

We evaluate our method on KITTI datasets (2012 and 2015) using the basic model which is trained on Flying Chairs. On KITTI2012, our method "Ours-S-600k" outperforms learning methods [5], [8], [28], [31], [33], [35], [37], [39] on both training and test sets. Traditional methods [1], [39] get results close to ours. Compared to [38], "Ours-S-600k" get a higher AEE on test set. However, the result of "Ours-C-1700k" is close to [38]. On KITTI2012 test set, "Ours-C-1700k" achieves the best result (AEE 5.77) in Table 2. We also tested "Ours-S-ft-600k (Sintel)" on KITTI2012 and KITTI2015 dataset. The experimental results show that the accuracy of the model fine-tuned on the Sintel dataset is slightly improved on the KITTI dataset. An example of optical flow estimation on KITTI2012 training dataset is shown in Fig. 6. On KITTI2015, [8], [28], [33] outperforms [5] and our models by a large margin. An example of optical flow estimation on KITTI2015 is shown in Fig. 7. On Middlebury, the AEE of our method is higher than [8], [27], and [28]. The results on KITTI and Middlebury datasets suggest that because our model is trained on simulated data, such as FlyingChairs, FlyingThings3D, the model is not adaptable enough for real-world scenarios such as KITTI and Middlebury datasets, and requires fine-tuning on different datasets.

3) FLYING CHAIRS

We split the Flying Chairs dataset into training set (22232 image pairs) and test set (640 image pairs). On test set, the results of "Ours-S-600k" is better than FlowNetS [5], but worse than FlowNetC [5]. The result of "Ours-C-1700k" is better than [5].

In summary, in Table 2, "Ours-S-600k" outperforms FlowNetS [5] on Sintel, KITTI, Middlebury and Flying Chairs datasets and gets best result on KITTI2012 training set. "Ours-C-1700k" gets the best performance on Sintel training and test set in Table 2. The experimental results demonstrate that adding prior assumptions into learning optical flow network can improve the accuracy of optical flow estimation. Finally, our model can product more refined and smoothed flow fields.

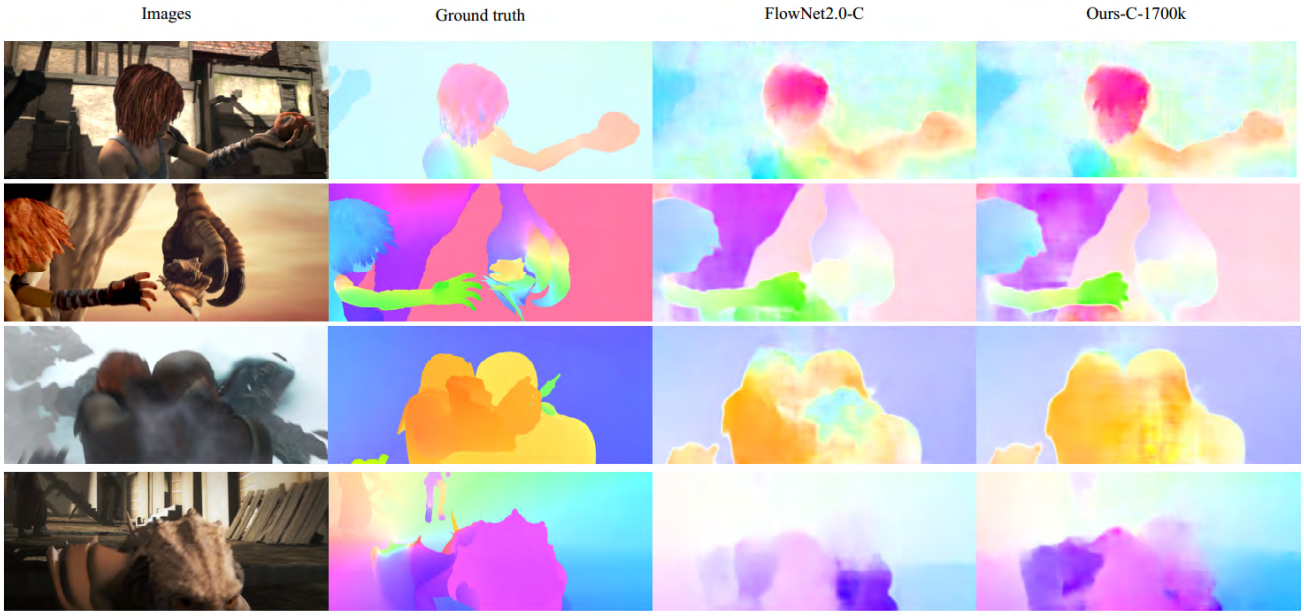


FIGURE 8. Examples of optical flow estimation from different methods using MPI-Sintel dataset (final version). In each row left to right: image, ground truth flow and 2 predictions: FlowNet2.0-C and "Ours-C-1700k" model. Note that our method can obtain more smoothed and detailed flow.

D. ABLATION STUDY

To understand the effects of each assumption in the proposed model, we conduct ablation analysis on different assumptions added in our model "Ours-S-600k". Table 4 shows the overall effects of them on MPI-Sintel training dataset. The final running time of the test process is shown the performance of "Ours-S-600k".

In Table 4, the top of two rows suggest that by adding brightness constancy assumption to the baseline network, the model improves it results from 4.50 to 4.37 on Sintel clean version and from 5.45 to 5.23 on Sintel final version. The gradient constancy assumption is also significant. The second and third rows show that adding gradient constancy assumption, the result of clean version improves from 4.37 to 4.33, and the result of final version improves from 5.23 to 5.21. As shown in the third and fourth rows, we find that adding the smoothness assumption (image-driven) also improves the performance of model on Sintel (clean and final versions). The bottom of row suggests that adding all assumptions together can get the best results on Sintel (clean and final versions). Visual results of optical flow estimation from different prior assumptions on Sintel final dataset are shown in Fig. 5.

E. RUNNING TIME

Running time is also an important factor for optical flow estimation. In Table 5, we first show the running time of other methods reported in their original experiments. Note that different methods were tested on different GPU models. Test environments for the original models are shown in Table 5. Variational methods [8], [13] were tested on CPU, and [8], [38], [39] were tested on GPU. All learning-based methods were only tested on GPU. Because our models used layers only implemented on GPU, we just tested our models on GPU. Further, we compared our models with traditional methods [38], [39] and learning-based method [5] on the

TABLE 5. Running time and average endpoint error on Sintel dataset.

Methods	CPU	GPU	Sintel final
EPPM [39] (NVIDIA GTX 780 GPU)	-	200ms	8.38
PCA-Flow [38] (NVIDIA Titan X GPU)	-	190ms	8.65
LDOF [8] (Intel Core 2.66GHz CPU, NVIDIA GTX Titan CPU)	65000ms	2500ms	9.12
Classic+NL [13] (Intel Core 3.0GHz CPU)	960000ms	-	9.15
FlowNetS [5] (NVIDIA GTX Titan GPU)	-	80ms	8.43
FlowNetC [5] (NVIDIA GTX Titan GPU)	-	150ms	8.81
DenseNetflow [35] (NVIDIA Titan X GPU)	-	130ms	10.07
SpyNet [28] (NVIDIA K80 GPU)	-	70ms	8.43
DSTFlow [37] (NVIDIA GTX Titan GPU)	-	80ms	11.11
RecSpyNet [31] (NVIDIA Titan X GPU)	-	70ms	9.38
EPPM [39] (NVIDIA 1080 Ti GPU)	-	164ms	8.38
PCA-Flow [38] (NVIDIA 1080 Ti GPU)	-	207ms	8.65
FlowNetS [5] (NVIDIA 1080 Ti GPU)	-	71ms	8.43
FlowNetC [5] (NVIDIA 1080 Ti GPU)	-	125ms	8.81
Ours-S-600k (NVIDIA 1080 Ti GPU)	-	71ms	8.19
Ours-C-600k (NVIDIA 1080 Ti GPU)	-	125ms	8.53

same GPU model (NVIDIA 1080 Ti). The results show that our models can improve the accuracy without increasing the running time and can obtain faster speed than traditional methods [38], [39]. Based on [5], we modified the loss function, which only works on training process. So, the running time of our model is same as [5]. In particular, our model is more accurate than traditional methods on Sintel final dataset.

V. CONCLUSION

We presented an end-to-end multi-assumption supervised CNNs for optical flow estimation. In our network, we leveraged prior assumptions from well-proven energy-based flow approaches, such as brightness constancy, gradient constancy and image-driven smoothness assumption. We showed that adding these assumptions during training can improve the accuracy of optical flow estimation and can guide network to get more detailed and smoothed flow fields. The experimental results on several benchmarks demonstrate the effectiveness of our framework. In future work, we will explore more robust loss function and novel network architecture.

REFERENCES

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

[2] 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, 2014.

[3] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 43–77, 1994.

[4] C. Zach, T. Pock, and H. Bischof, "A duality based approach for real-time TV-L^1 optical flow," in *Pattern Recognition*. Berlin, Germany: Springer-Verlag, 2007, pp. 214–223.

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

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

[7] 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. (ECCV)*, May 2004, pp. 25–36.

[8] 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.

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

[10] 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.

[11] 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.

[12] H.-H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 5, pp. 565–593, Sep. 1986.

[13] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2432–2439.

[14] M. Hua, X. Bie, M. Zhang, and W. Wang, "Edge-aware gradient domain optimization framework for image filtering by local propagation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 2838–2845.

[15] Z. Tu, W. Xie, J. Cao, C. van Gemeren, R. Poppe, and R. C. Veltkamp, "Variational method for joint optical flow estimation and edge-aware image restoration," *Pattern Recognit.*, vol. 65, pp. 11–25, May 2017.

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

[17] B. Banerjee and V. Murino, "Efficient pooling of image based CNN features for action recognition in videos," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2637–2641.

[18] Z. Gao, L. Wang, L. Zhou, and J. Zhang, "HEp-2 cell image classification with deep convolutional neural networks," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 2, pp. 416–428, Mar. 2017.

[19] M. Ravanbakhsh, H. Mousavi, M. Nabi, M. Rastegari, and C. Regazzoni, "CNN-aware binary MAP for general semantic segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1923–1927.

[20] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: A survey," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 84–100, Jan. 2018.

[21] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.

[22] D. Sun, S. Roth, J. P. Lewis, and M. J. Black, "Learning optical flow," in *Proc. 10th Eur. Conf. Comput. Vis. III (ECCV)*, 2008, pp. 83–97.

[23] Y. Li and D. P. Huttenlocher, "Learning for optical flow using stochastic optimization," in *Computer Vision—ECCV*, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Germany: Springer, 2008, pp. 379–391.

[24] D. Rosenbaum, D. Zoran, and Y. Weiss, "Learning the local statistics of optical flow," in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, Inc, 2013, pp. 2373–2381.

[25] V. Vaquero, G. Ros, F. Moreno-Noguer, A. M. Lopez, and A. Sanfeliu, "Joint coarse-and-fine reasoning for deep optical flow," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2558–2562.

[26] N. Mayer *et al.*, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4040–4048.

[27] D. Teney and M. Hebert, "Learning to extract motion from videos in convolutional neural networks," in *Computer Vision—ACCV*. Cham, Switzerland: Springer, 2016, pp. 412–428.

[28] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2720–2729.

[29] 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)*, Jul. 2017, pp. 1647–1655.

[30] D. Zhang, G. Guo, D. Huang, and J. Han, "PoseFlow: A deep motion representation for understanding human behaviors in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 6762–6770.

[31] P. Hu, G. Wang, and Y.-P. Tan, "Recurrent spatial pyramid CNN for optical flow estimation," *IEEE Trans. Multimedia*, to be published.

[32] A. Ahmadi and I. Patras, "Unsupervised convolutional neural networks for motion estimation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1629–1633.

[33] J. J. Yu, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *Proc. Workshops Comput. Vis. (ECCV)*, 2016, pp. 3–10.

[34] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu, "Learning image matching by simply watching video," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2016, pp. 434–450.

[35] Y. Zhu and S. Newsam, "DenseNet for dense flow," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 790–794.

[36] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[37] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, "Unsupervised deep learning for optical flow estimation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2017, p. 7.

[38] J. Wulff and M. J. Black, "Efficient sparse-to-dense optical flow estimation using a learned basis and layers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 120–130.

[39] L. Bao, Q. Yang, and H. Jin, "Fast edge-preserving patchmatch for large displacement optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3534–3541.
[40] 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. (ECCV)*, Oct. 2012, pp. 611–625.

XUEZHI XIANG was born in Harbin, China, in 1979. He received the B.Eng. degree in information engineering, and the M.Sc. and Ph.D. degrees in signal and information processing from Harbin Engineering University, China, in 2002, 2004, and 2008, respectively. He was a Post-Doctoral Fellow with the Harbin Institute of Technology from 2009 to 2011. Since 2010, he has been an Associate Professor with the School of Information and Communication Engineering, Harbin Engineering University. From 2011 to 2012, he was a Visiting Scholar with the University of Ottawa. He has authored over 40 articles. His research interests include image processing, computer vision, and pattern recognition. He is a member of the Association for Computing Machinery and a Senior Member of the China Computer Federation.

MINGLIANG ZHAI was born in Xining, China, in 1994. He received the B.Eng. degree from Jilin University, China, in 2016. His research interests include image processing, computer vision, and pattern recognition.

RONGFANG ZHANG was born in Daqing, China, in 1993. She received the B.Eng. degree in communication engineering from Harbin Engineering University, China, in 2017. Her research interests include image processing, computer vision, and pattern recognition.

YULONG QIAO received the B.S. and M.S. degrees in applied mathematics from the Harbin Institute of Technology (HIT), Harbin, China, in 2000 and 2002, respectively, and the Ph.D. degree from the Department of Automatic Test and Control, HIT, in 2006. He is currently a Professor with the School of Information and Communications Engineering, Harbin Engineering University. His current research interests include wavelet theory, image and video processing, texture analysis, and pattern recognition.

ABDULMOTALEB EL SADDIK (F'09) is currently a Distinguished University Professor and the University Research Chair with the School of Electrical Engineering and Computer Science, University of Ottawa. He has authored and co-authored four books and over 550 publications and chaired over 50 conferences and workshop. His research focus is on multimodal interactions with sensory information in smart cities. He has received research grants and contracts totaling over 18 M. He has supervised over 120 researchers and received several international awards, among others, are an ACM Distinguished Scientist, a fellow of the Engineering Institute of Canada, and the Canadian Academy of Engineers, the IEEE I&M Technical Achievement Award and the IEEE Canada Computer Medal. He is a Senior Associate Editor among others of the *ACM Transactions on Multimedia Computing, Communications, and Applications* (ACM TOMCCAP currently TOMM), the IEEE TRANSACTIONS ON MULTIMEDIA, and the guest editor of several IEEE TRANSACTIONS and journals.