OIFlow: Occlusion-Inpainting Optical Flow Estimation by Unsupervised Learning

Shuaicheng Liu, *Member, IEEE*, Kunming Luo, Nianjin Ye, Chuan Wang, *Member, IEEE*, Jue Wang, *Senior Member, IEEE*, and Bing Zeng, *Fellow, IEEE*

***Abstract*— Occlusion is an inevitable and critical problem in unsupervised optical flow learning. Existing methods either treat occlusions equally as non-occluded regions or simply remove them to avoid incorrectness. However, the occlusion regions can provide effective information for optical flow learning. In this paper, we present OIFlow, an occlusion-inpainting framework to make full use of occlusion regions. Specifically, a new appearance-flow network is proposed to inpaint occluded flows based on the image content. Moreover, a boundary dilated warp is proposed to deal with occlusions caused by displacement beyond the image border. We conduct experiments on multiple leading flow benchmark datasets such as Flying Chairs, KITTI and MPI-Sintel, which demonstrate that the performance is significantly improved by our proposed occlusion handling framework.**

***Index Terms*— Optical flow, inpainting, appearance flow, occlusion, unsupervised learning.**

I. Introduction

OPTICAL flow estimation is a fundamental vision task, which has been widely used in many applications such as action recognition [1], video analysis [2]–[4], and automatic driving [5]. There are various challenges regarding accurate flow estimation, such as large displacements [6], noisy interferences [7], motion details [8], [9] and occlusions [10]. Traditional approaches optimize hand-crafted energy functions to obtain flow fields [11]–[15]. Learning-based methods, on the other hand, directly estimate flows from images by convolutional neural networks (CNNs) [16]. Various network designs have been proposed to overcome these problem-inherit challenges [17]–[21] to improve the performance.

Among all these challenges, the most difficult one is the occlusion, which is the key issue that prevents high-quality performance and practical applications. Occlusion comes from dynamic objects, discontinuous depth disparities, and frame boundaries. Traditional methods often regularize incoherent motion to propagate flow from non-occluded surroundings to occluded parts [22], [23]. However, this could fail when occlusion regions are wrongly located or motions are propagated from incorrect motion sources. Whereas, training supervised CNNs requires a large amount of annotated training data [24], which is hard to obtain in practice, especially when there are occlusions [25], [26]. In addition, supervised networks can only be trained on the synthetic data due to the absence of ground-truth labels of the real images, leading to the domain adaptation problem in real applications.

On the other hand, unsupervised optical flow can learn from unlabeled data by minimizing photometric loss between images, such that the training data is no longer a bottleneck. However, directly minimizing photometric loss cannot learn correct flows in occlusion regions due to incorrect warped results. Unfortunately, early works of the unsupervised family simply ignored the influence of occlusions and treated them equally as non-occlusion regions [27]–[29]. To handle this, Wang *et al.* [30] proposed an occlusion aware learning where occlusions are detected and discarded during loss calculation. DDFlow [31] and SelFlow [32] applied random cropping to hallucinate occlusions for self-supervisions. However, all these methods miss the guidance in real occlusion regions.

To this end, we propose our OIFlow which not only locates but also inpaints real occlusions. Specifically, we treat the error-prone occlusion regions as missing regions and inpaint these regions by appearance flow, which is originally used to propagate pixels or features from the source region to the target region in the tasks of view synthesis [33] and image inpainting [34]. Here, we modified the appearance flow network that was proposed for the task of image inpainting [34] to inpaint missing flows contextually. Fig. 1 demonstrates our motivation. Fig. 1 (a), (b) and (e) show examples of reference frame, ground truth flow and detected object occlusion mask, respectively. Fig. 1 (c) simply discards occluded regions during training. The flow on the back of the chair is incorrect. Fig. 1 (d) is our result with occlusion regions inpainted by our appearance flow refinement block. The motion of the chair is now correct. Specifically, we show a zoom-in region in Fig. 1 (f), where the inpainting is applied on the image domain to inpaint from non-occlusion regions to occlusion regions according to the image contents (red and blue arrows). The same inpainting procedure is copied to the flow field (Fig. 1 (g)). Fig. 1 (h) shows the inpainted flow field.

Manuscript received November 1, 2020; revised May 18, 2021 and June 23, 2021; accepted June 23, 2021. Date of publication July 7, 2021; date of current version July 15, 2021. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61872067, Grant 62031009, and Grant 61720106004; in part by the 111 Projects under Grant B17008; and in part by the Sichuan Science and Technology Program under Grant 2019YFH0016. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zhen He. *(Corresponding author: Bing Zeng.)*

Shuaicheng Liu and Bing Zeng are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: eezeng@uestc.edu.cn).

Kunming Luo, Nianjin Ye, Chuan Wang, and Jue Wang are with Megvii Technology, Chengdu 610095, China.

Digital Object Identifier 10.1109/TIP.2021.3093781

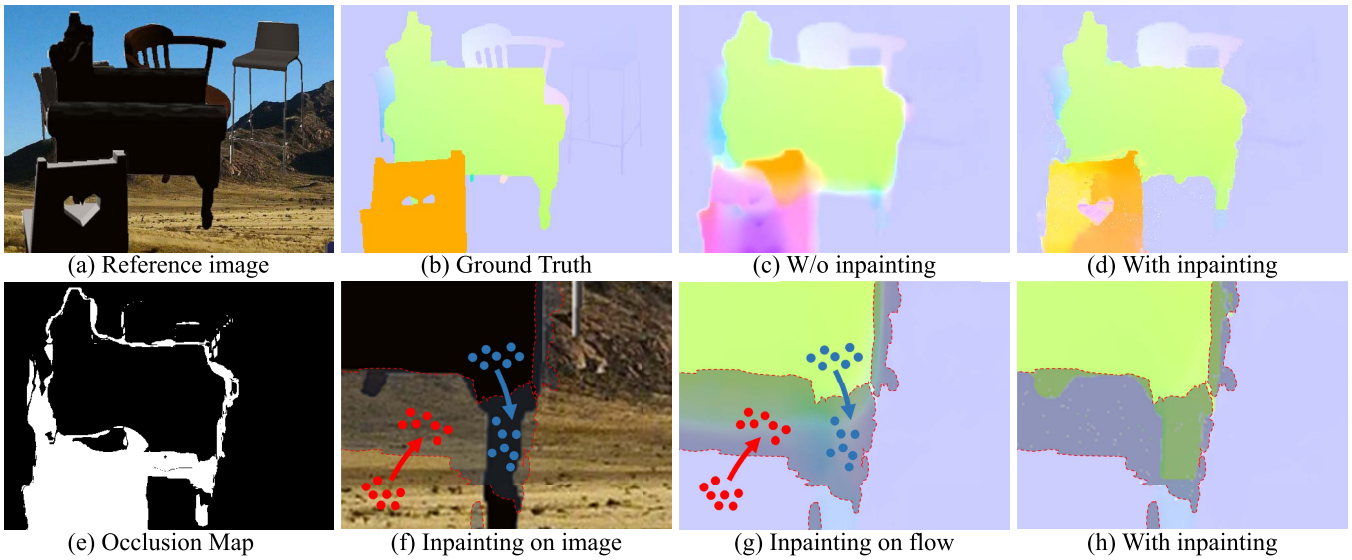


Fig. 1. (a) reference image. (b) ground truth flow. (c)(d) results without / with the optical flow refinement. (e) detected occlusion map. (f) a zoom-in window with mask (dark region) overlaid on the image. Here an appearance flow is learned on the image domain, which is further used to inpaint the occluded regions by the non-occluded ones. (g) the appearance flow of (f) is applied to guide the propagation in the flow field. (h) The propagated results in the zoom-in region.

In addition, we further divide occlusions into two types, object occlusion and boundary occlusion. For the former one, which occurs within the image, we inpaint its flow according to contextual relevance. For the latter one, which is caused by displacement beyond the image boundary, we propose a boundary dilated warp to correct its loss. Moreover, our network backbone is built upon the supervised framework IRR-PWC [35] by absorbing its merits but changed from supervised to the unsupervised setting.

Through extensive experiments on multiple representative datasets, we demonstrate that our OIFlow can significantly improve the performance, exceeding all existing unsupervised methods. In particular, on MPI-Sintel benchmark test, we achieve EPE $= 4.26$ on the Clean pass (the second best method is 4.78) and EPE $= 5.71$ on the Final pass (the second best method is 5.89), which even outperform some supervised methods. Our code and trained models will be available online. Our contributions are summarized as follows:

- An occlusion inpainting framework named **OIFlow** is proposed for unsupervised optical flow estimation.
- A new appearance-flow network named **AppFlowNet-Lite** is proposed to inpaint occluded flows based on the image content to provide additional guidance in occlusions. Such important regions are either treated as non-occlusions or totally abandoned previously.
- A **Boundary Dilated Warping** is proposed to handle pixels caused by frame boundary occlusions, which is effective and not attempted before.

II. Related Work

A. Supervised Deep Optical Flow

Supervised methods learn optical flow from annotated data [36]–[39]. FlowNet [16] is the first work to estimate optical flow using a fully convolutional network. The following work FlowNet2 [17] extended FlowNet by stacking networks iteratively, which largely improved the performance. For large displacements, SPyNet [20] proposed to warp images based on the coarse-to-fine manner. Then, PWCNet [40], [41] and LiteFlowNet [21], [42] calculated cost volume after warping features, resulting in efficient and light-weight networks. IRRPWC [35] proposed an iterative residual refinement scheme for optical flow network design. Recently, in order to extract better correspondence information for optical flow estimation, VCN [39] built 4D volumetric correspondence for capturing complementary notions of match cost. MaskFlowNet [37] proposed to filter out useless areas during building the cost volume. DICL-Flow [43] learned matching costs from concatenated features with a displacement-invariant cost learning module. More recently, LiteFlowNet3 [44] improved the performance of optical flow estimation by cost volume modulation and flow field deformation. RAFT [45] proposed to build 4D correlation volume and estimate optical flow by the recurrent network, yielding state-of-the-art performance on multiple benchmarks. However, The performance of these deep learning-based methods depend heavily on the training data, which is prohibitively expensive in practical applications.

B. Unsupervised Deep Optical Flow

Unsupervised methods do not require flow annotations [27]–[29], where the photometric loss is optimized during training. However, the photometric loss cannot work in occlusion regions and even lead to incorrect guidance. Therefore, works [30], [46] proposed to exclude occlusion regions from photometric loss for improvements.

Recently, researchers introduced multi-frame formulation [47], epipolar constrain [48], depth estimation

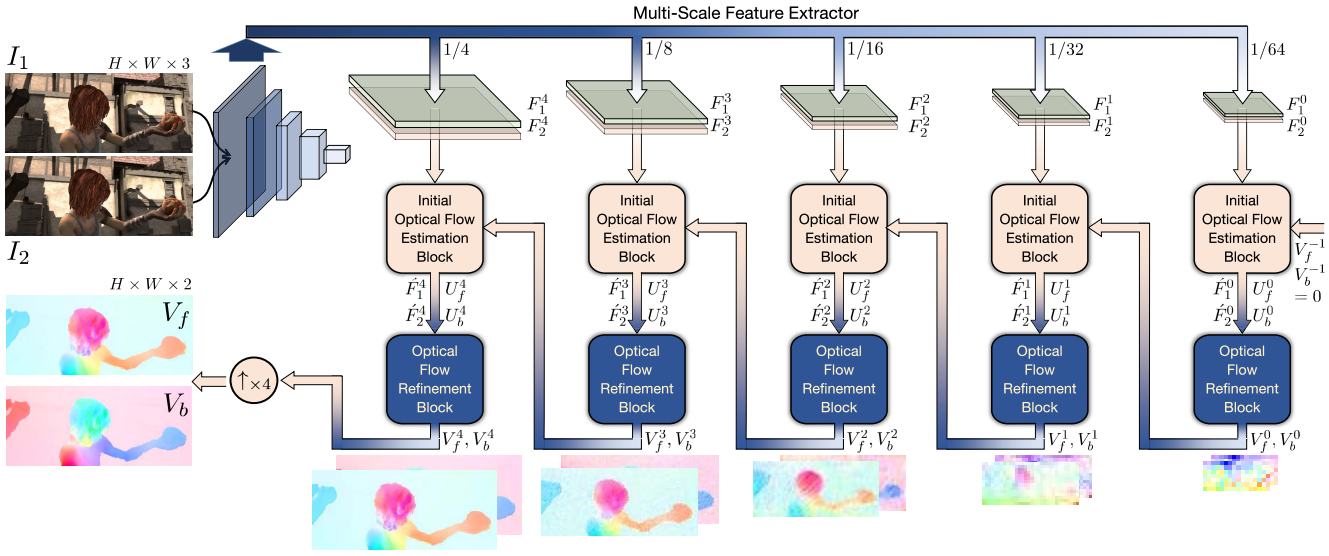


Fig. 2. The network structure of our OIFlow Network, which involves 3 main parts, a multi-scale feature extractor, and an initial optical flow estimation block (Fig. 3) together with a refinement block (Fig. 4) working on each scale.

[49]–[52], stereo matching [53], feature similarity constrain [54] and non-local term [55] to further constraint the problem. Concurrently, Back2Future [47] proposed to take advantage of the guidance information in multiple frames for optical flow learning. EpiFlow [48] introduced other guidance information such as the low-rank information and the sub-space information. PatchFlow [56] calculated photometric loss by patch-based warping and estimated occlusion mask by a CNN-based branch. STFlow [57] proposed an iterative self-taught framework in which the traditional flow interpolation method [23] is used to improve the self-estimated flow to provide pseudo labels for network training. DDFlow [31] and SelFlow [32] proposed to learn optical flow based on data distillation. They artificially create occlusions for self supervision. Recently, ARFlow [58] proposed to learn optical flow from image sequences with the self-supervision from augmentations. UFlow [59] proposed to analyze different unsupervised components in a unified framework systematically, which achieved state-of-the-art performance. Although the performance can be largely improved by previous methods, the real occlusions still have no guidance.

In this paper, we provide guidance in occlusion regions by our appearance flow warp and boundary warp.

C. Appearance Flow

Appearance flow is first proposed by [33] for view synthesis. Appearance flow can copy pixels from source images for target object generation, which is superior to generating from scratch. Recently, appearance flow is applied for image inpainting [34], where the appearance flow can propagate features from existing regions to missing regions for texture generation. In this paper, we modify the appearance flow for optical flow inpainting.

III. Our Method

A. Network Structure

Our method is built upon a supervised optical flow estimation framework IRR-PWC [35], which is further adapted to enable unsupervised learning and occlusion awareness. It takes two frames I_1 and I_2 of size $H \times W \times 3$ as input and produces forward and backward optical flows V_f, V_b of size $H \times W \times 2$ as output. The entire network can be divided into three parts, i.e., a feature extractor that extracts multi-scale feature maps of inputs, an initial optical flow estimation block and a newly introduced optical flow refinement block. We illustrate the network structure in Fig. 2.

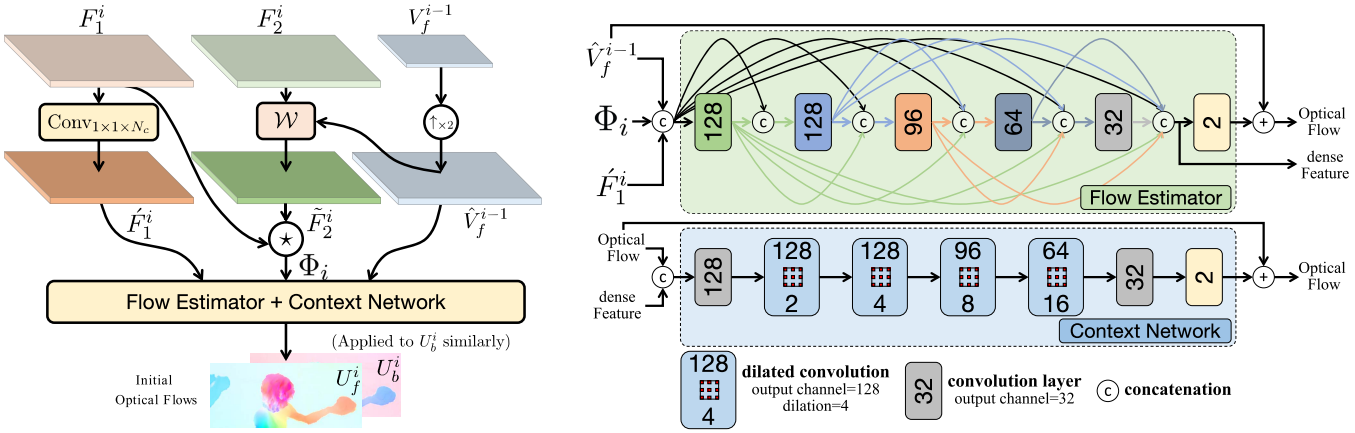
1) Multi-Scale Feature Extractor: Given two consecutive frames I_1, I_2, we first extract $(N+1)$ scales of feature maps $\{F_k^i\}, i = 0, 1, \ldots, N$ for each I_k $(k \in \{1, 2\})$ separately, by a fully convolutional encoder $\mathcal{E}$. For each frame, its successive feature maps F_k^i and $F_k^{i-1} (k \in \{1, 2\})$ follow a relationship of $2\times$ scale. In our design, we denote the feature map of $1/64$ image size as the 0-th one, so that the i-th feature map is of 2^{i-6} image size, $i = 0, 1, \ldots, N$. As such, the feature maps extracted are written as follows.

$$\left\{F_k^i\right\} = \mathcal{E}(I_k), \quad \text{for } i \in \{0, 1, \ldots, N\} \text{ and } k \in \{1, 2\} \quad (1)$$

In practice, we set N to 4 so that the finest scaled feature maps extracted by $\mathcal{E}$ are of $1/4$ image size. This size is a balanced result considering the accuracy and efficiency.

Note that the structure of the multi-scale feature extractor is the same as IRR-PWC [35] where 6 convolution blocks are used to extract feature maps in multiple scales. Each convolution block contains 2 convolutional layers where one layer with stride 2 to downscale the feature map and another layer with stride 1 to translate feature dimensions.

2) Initial Optical Flow Estimation Block: We further generate *initial* forward and backward optical flows for each scale in a coarse-to-fine manner. For the i-th scale, the feature pair



(a) Initial Optical Flow Estimation Block. (b) Flow Estimator and Context Network.

Fig. 3. The structure of the initial optical flow estimation block. (a): the pipeline of the initial optical flow estimation block. (b): the detailed structures of the flow estimator and the context network.

(F_1^i, F_2^i) is further fed into an estimator $\mathcal{D}$, together with the *refined* optical flows in the $(i-1)$-th scale, i.e. V_f^{i-1}, V_b^{i-1}. The estimator $\mathcal{D}$ produces two initial optical flows U_f^i, U_b^i as:

$$U_f^i = \mathcal{D}\big(F_1^i, F_2^i, V_f^{i-1}\big), \quad U_b^i = \mathcal{D}\big(F_2^i, F_1^i, V_b^{i-1}\big), \tag{2}$$

Specifically, the feature maps $\{F_1^i\}$ from multiple scales may have various numbers of channels, we add one 1×1 convolutional layer following F_1^i, to get a feature map $\acute{F}_1^i$ of a uniform number of channels N_c (Eq. 3). Second, as the refined optical flow V_f^{i-1} from the $(i-1)$-th scale is only $1/2$ size of F_k^i, we upsample it to obtain its $2\times$ version $\hat{V}_f^{i-1}$. Note that in all the flow field upsampling processes, we multiply the flow field by the corresponding upsample factor to represent the correct motion in the higher resolution. After the flow upsampling, we further applied $\hat{V}_f^{i-1}$ to warp F_2^i to obtain $\tilde{F}_2^i$ (Eq. 4). Then, we calculate the correlation of F_1^i and $\tilde{F}_2^i$ (Eq. 5), and finally feed the three tensors to a flow estimator followed by a context network, to obtain the initial forward optical flow U_f^i (Eq. 6). To sum up, the process above can be described by the following equation:

$$\acute{F}_1^i = \mathrm{Conv}_{1\times1\times N_c}(F_1^i), \tag{3}$$

$$\hat{V}_f^{i-1} = \mathrm{Sample}_{\uparrow\times2}(2V_f^{i-1}), \quad \tilde{F}_2^i = \mathcal{W}(F_2^i, \hat{V}_f^{i-1}), \tag{4}$$

$$\Phi_i = F_1^i \star \tilde{F}_2^i, \tag{5}$$

$$U_f^i = \mathcal{D}\big(F_1^i, F_2^i, V_f^{i-1}\big) = \mathcal{H}(\acute{F}_1^i, \Phi_i, \hat{V}_f^{i-1}), \tag{6}$$

where $\mathcal{W}$ is a conventional warping operation achieved by pixel re-mapping, $\star$ is the correlation operator and $\mathcal{H}$ represents the combination of the flow estimator and context network. Similarly, we obtain the backward optical flow U_b^i by swapping F_1^i, F_2^i and replacing V_f^{i-1} by V_b^{i-1}, i.e. $U_b^i = \mathcal{D}(F_2^i, F_1^i, V_b^{i-1})$. Specifically, we set $V_f^{-1} = V_b^{-1} = 0$ for scale $i = 0$.

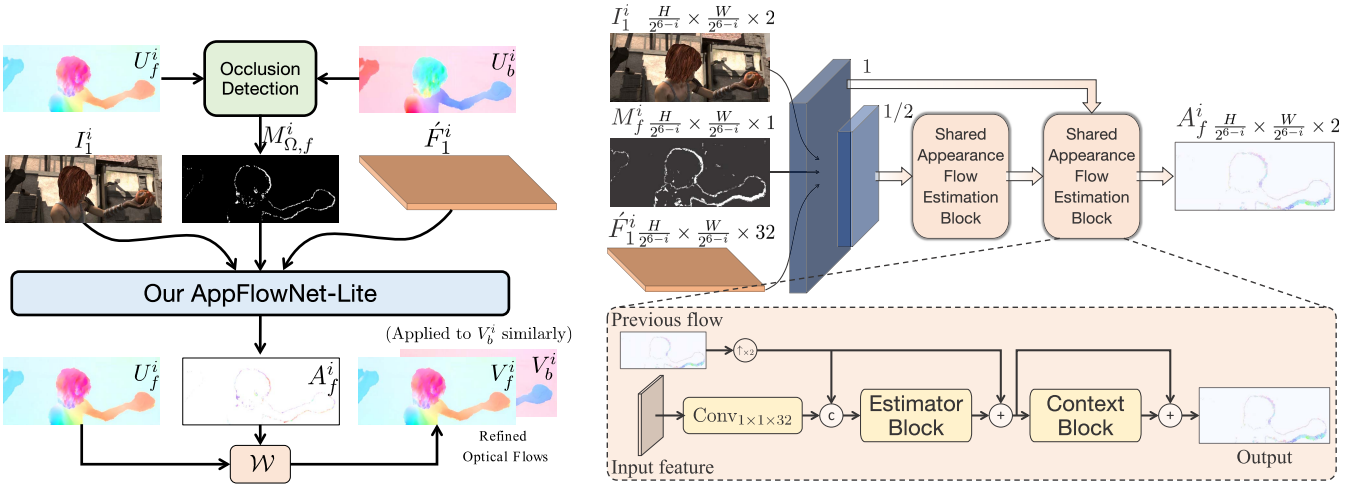
The structure of $\mathcal{D}$ is shown in Fig. 3(a), where a flow estimator and a context network are used to estimate the initial optical flow after feature warping and correlation. The architecture of the flow estimator and the context network are the same as that in IRR-PWC [35]. As detailed in Fig. 3(b), a dense block with 6 convolutional layers is used in the flow estimator to produce residual optical flow and a dense feature. Then, in the context network, 4 dilated convolution layers and 3 conventional convolutional layers are used to extract context information from the dense feature to refine optical flow.

3) Optical Flow Refinement Block: The initial optical flows estimated above commonly have incorrect pixels caused by occlusions. Severe artifacts may occur if we directly apply warping. We propose to first detect an occlusion map and then to inpaint the occluded regions using the optical flow in the non-occluded ones. In our framework, we obtain the occlusion map by comparing the initial forward and backward optical flows U_f^i, U_b^i, and design a light version of the Appearance Flow Network [34] (AppFlowNet) as our inpainting approach, named **AppFlowNet-Lite** (Detailed in Sec. III-A4). Specifically, we follow the forward-backward consistency prior of optical flow as in [46] to check whether a pixel is still within its neighborhood after being moved by the forward and backward flow vectors,

$$M_{\Omega,f}^i(\boldsymbol{p}) = \begin{cases} 1, & \text{if } \big|V_f^i(\boldsymbol{p}) + V_b^i\big(\boldsymbol{p} + V_f^i(\boldsymbol{p})\big)\big| \\ & \quad > \alpha_1\big(|V_f^i(\boldsymbol{p})| + |V_b^i(\boldsymbol{p} + V_f^i(\boldsymbol{p}))|\big) + \alpha_2 \\ & \quad \text{and } \boldsymbol{p} + V_f^i(\boldsymbol{p}) \in \Omega \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

where $\boldsymbol{p}$ is a pixel coordinate, α_1, α_2 are threshold hyper-parameters, Ω is the image plane, and $M_{\Omega,f}^i$ is the occlusion map. In the occlusion map $M_{\Omega,f}^i$, '1' represents the occlusion pixels within the image plane and '0' for other pixels (non-occlusion pixels and the out-of-view occlusion pixels). Note that we only refine the in-frame occlusion regions in our optical flow refinement block, because the out-of-view occlusion regions can be learned by properly modifying the photometric loss using our boundary dilated warping, which is detailed in Sec. III-B1.

After $M_{\Omega,f}^i$ is acquired, the normalized feature $\acute{F}_1^i$, the downsampled image I_1^i and the occlusion mask $M_{\Omega,f}^i$ are used as the input of the AppFlowNet-Lite $\mathcal{A}$ to produce an appearance flow A_f^i. In A_f^i, the matching vectors are



(a) Optical Flow Refinement Block. (b) Framework of AppFlowNet-Lite.

Fig. 4. The structure of the optical flow refinement block. (a): the pipeline of the optical flow refinement block. (b): the detailed structure of the AppFlowNet-Lite.

generated for the pixels in occlusions, so that by warping U_f^i with A_f^i, a refined flow V_f^i is achieved. Likewise, V_b^i, A_b^i and $M_{\Omega,b}^i$ can be obtained similarly. In summary,

$$V_\beta^i = \mathcal{W}(U_\beta^i, A_\beta^i), \quad A_\beta^i = \mathcal{A}(\acute{F}_k^i, I_k^i, M_{\Omega,\beta}^i), \tag{8}$$

where $\langle k, \beta\rangle \in \{\langle 1, f\rangle, \langle 2, b\rangle\}$, I_k^i are the corresponding down-sampled version of $I_k, k \in \{1, 2\}$ in i-th scale. Note that here we also feed $\mathcal{A}$ with $\acute{F}_k^i$ just for avoiding redundant feature extraction steps and saving network parameters. See Fig. 4(a) for the illustration of this block.

4) Our AppFlowNet-Lite: The original AppFlowNet [34] follows a classical Encoder-Decoder framework and involves a moderate scale of parameters. To facilitate the efficiency of our framework, we follow the similar idea and design a light version of it. We remove several layers from the original network to make our implementation handle only 2 scales of feature maps instead of 4 scales. Also, we make the parameters of the two decoding layers shared so that more weights can be saved. With these schemes, our AppFlowNet-Lite contains only **3.82**% of the parameters of the original one (2.06 M vs. 53.86 M), but the performance is demonstrated stable enough.

The structure of our AppFlowNet-Lite is shown in Fig. 4(b). In the entire network, we first use several convolutional layers to extract two feature maps in 1 and $1/2$ scale. Then, a shared decoder is utilized to produce the appearance flow. The detailed structure of the shared decoder is illustrated in Fig. 4(b), where we first use a 1×1 convolutional layer to modify the number of the input feature channels into a uniform one. Then we concatenate the unified feature with the previous appearance flow map and feed it into an estimator block to produce a new appearance flow. Note that the initial flow map for the decoder in $1/2$ scale is set as zeros. Also, in order to learn residual information, a context block is connected following the Estimator Block to better refine the appearance flow. As for the architecture of the Estimator and Context Blocks, we simply use 5 densely connected convolutional layers in the former one, 5 dilated convolution layers with dilation $\{1, 2, 4, 8, 16\}$ and two convolutional layers in the latter one to produce the residual information.

With the scale becoming finer, we obtain higher quality flows with occlusions well inpainted. After we finish the N-th scale with V_f^N and V_b^N, we directly upsample them to the size of $H \times W \times 2$, and apply the refinement block to them. By doing this, V_f, V_b are generated as the final output.

B. Unsupervised Training

We customize the unsupervised losses to characterize the pixel alignment. Due to the existence of two types of occlusion, that are caused by internal objects and the limited image plane, we design corresponding warping strategies. For the former one, we propose to utilize the standard warping operation $\mathcal{W}$ based on our learned appearance flow, while for the latter case, we introduce a Boundary Dilated Warping before formulating our unsupervised loss.

1) Boundary Dilated Warping: As aforementioned, some of the pixels where $M_{\Omega,\beta}^i(\boldsymbol{p}) = 0$ may be moved outside the image plane. This type of pixels usually stay close to the four external image boundaries. In order to make them directly trainable in the unsupervised setting, we propose a boundary dilated warping strategy $\mathcal{W}_\Omega$. Fig. 5 shows an example. The original reference image I_1^r and target image I_2^r are cropped into the training image pair I_1 and I_2. Our goal is to warp the target image I_2 based on the optical flow V_f. The traditional warp method $\mathcal{W}$ directly warps pixels of I_2 by V_f, ignoring motions toward the outside of the image plane. Thus the upper region of the warped image is filled with zeros, i.e. black pixels. To solve the problem, we design the boundary dilated warping method that warps pixels of I_2^r based on V_f, and pixels moving outside the image plane of I_2 can be mapped back and used to fill in the missing regions.

Specifically, in the training stage, we feed the network with a cropped version of the raw image pair I_1^r, I_2^r. The cropped

TABLE I

COMPARISON WITH EXISTING METHODS. OUR METHOD OUTPERFORMS ALL THE UNSUPERVISED OPTICAL FLOW METHODS ON KITTI AND SINTEL DATASETS. WE REPORT EPE (THE LOWER THE BETTER) VALUE AS RESULTS ON ALL THE BENCHMARKS EXCEPT THE TEST SET OF KITTI 2015 WHERE F1 (THE LOWER THE BETTER) MEASUREMENT IS USED BY FOLLOWING THE ONLINE EVALUATION. '–' INDICATES THE RESULTS ARE NOT REPORTED, (·) INDICATES THE TESTING IS CONDUCTED ON THE TRAINING DATASET, AND '-SYN' MEANS THE MODEL IS ONLY TRAINED ON SYNTHETIC DATASETS

	Method	Chairs	KITTI 2012			KITTI 2015			Sintel Clean		Sintel Final	
		test	train	test-all	test-noc	train	test-all	test-noc	train	test	train	test
Supervised	FlowNetS-syn [16]	2.71	8.26	–	–	–	–	–	4.50	7.42	5.45	8.43
	FlowNetS [16]	–	7.52	9.1	5.0	–	–	–	(3.66)	6.96	(4.44)	7.76
	SpyNet-syn [20]	2.63	9.12	–	–	–	–	–	4.12	6.69	5.57	8.43
	SpyNet [20]	–	8.25	4.1	2.0	–	35.07%	26.71%	(3.17)	6.64	(4.32)	8.36
	FlowNet2-syn [17]	–	4.09	–	–	10.06	–	–	2.02	3.96	3.14	6.02
	FlowNet2 [17]	–	(1.28)	1.8	1.0	(2.3)	10.41%	6.94%	(1.45)	4.16	(2.01)	5.74
	PWC-Net-syn [40]	2.00	4.57	–	–	13.20	–	–	3.33	–	4.59	–
	PWC-Net [40]	–	(1.45)	1.7	0.9	(2.16)	9.60%	6.12%	(1.70)	3.86	(2.21)	5.13
	IRR-PWC [35]	1.67	–	1.6	0.8	(1.63)	7.65%	4.86%	(1.92)	3.84	(2.51)	4.58
Unsupervised	BackToBasic [29]	5.3	11.3	9.9	4.6	–	–	–	–	–	–	–
	DSTFlow [28]	5.11	10.43	12.4	4.0	16.79	39%	–	(6.16)	10.41	(6.81)	11.27
	UnFlow [47]	–	3.29	–	–	8.10	23.3%	–	–	9.38	(7.91)	10.22
	OAFlow-syn [30]	3.30	12.95	–	–	21.30	–	–	5.23	8.02	6.34	9.08
	OAFlow [30]	–	3.55	4.2	–	8.88	31.2%	–	(4.03)	7.95	(5.95)	9.15
	PatchFlow [57]	3.65	3.34	4.0	–	6.91	21.82%	–	(4.45)	7.70	(4.99)	7.98
	Back2Future [48]	–	–	–	–	6.59	22.94%	13.85%	(3.89)	7.23	(5.52)	8.81
	NLFlow [56]	2.98	3.02	4.5	–	6.05	22.75%	–	(2.58)	7.12	(3.85)	8.51
	DDFlow-syn [31]	2.97	8.27	–	–	17.26	–	–	3.83	–	4.85	–
	DDFlow [31]	–	2.35	3.0	1.1	5.72	14.29%	9.55%	(2.92)	6.18	(3.98)	7.40
	EpiFlow [49]	–	(2.51)	3.4	–	(5.55)	16.95%	–	(3.54)	7.00	(4.99)	8.51
	SelFlow [32]	–	1.69	2.2	1.0	4.84	14.19%	9.65%	(2.88)	6.56	(3.87)	6.57
	STFlow [56]	**2.53**	1.64	1.9	**0.9**	3.56	13.83%	9.70%	(2.91)	6.12	(3.59)	6.63
	ARFlow [59]	–	1.44	1.8	–	2.85	11.80%	–	(2.79)	4.78	(3.87)	5.89
	UFlow [60]	2.55	1.68	1.9	**0.9**	2.71	11.13%	8.41%	(2.50)	5.21	(3.39)	6.50
	Ours	**2.53**	**1.33**	**1.6**	1.0	**2.57**	**9.81%**	**7.73%**	(**2.44**)	**4.26**	(**3.35**)	**5.71**

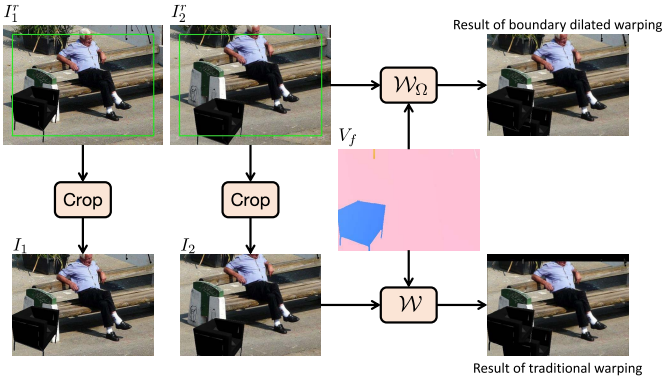


Fig. 5. An example of the boundary dilated warp (upper row) vs. the traditional warp (lower row). The traditional warp often introduces empty regions.

images I_1, I_2 are achieved as follows,

$$I_k(\boldsymbol{p}) = I_k^r(\boldsymbol{p}_0 + \boldsymbol{p}), \quad k \in \{1, 2\} \tag{9}$$

where $\boldsymbol{p}_0$ is the upper-left coordinate of the cropped region in $I_k^r, k \in \{1, 2\}$, and we need to keep I_k is totally covered by I_k^r. It looks like $\{I_k^r\}$ are the "dilated" version of $\{I_k\}$. As such, our boundary dilated warping is expressed as below,

$$\tilde{I}_k(\boldsymbol{p}) = \mathcal{W}_\Omega(I_k^r, V_\beta, \boldsymbol{p}_0)(\boldsymbol{p}) = I_k^r\big(\boldsymbol{p}_0 + \boldsymbol{p} + V_\beta(\boldsymbol{p})\big), \tag{10}$$

where $\langle k, \beta\rangle \in \{\langle 2, f\rangle, \langle 1, b\rangle\}$.

Recently, DDFlow [31] proposed a two-stage data distillation method to learn optical flow in boundary occlusion regions. They first artificially produce out-of-view occlusion regions by random cropping. Then they use the predictions of a pre-trained teacher model as pseudo labels to stimulate the student model learning to estimate optical flow in these newly occluded regions. In contrast, our boundary dilated warping is different from DDFlow because it is a one-stage method. Specifically, we use boundary dilated warping to modify the alignment losses so that they can provide correct guidance information for optical flow learning in the boundary occlusion areas. Therefore, networks can correctly estimate the optical flow in out-of-view occlusion regions after end-to-end unsupervised learning with our boundary dilated warping enabled.

2) Unsupervised Losses: In our framework, we use three losses: the alignment loss for non-occlusion regions and boundary occlusion regions, the occlusion inpainting loss for in-frame occlusion regions, the smoothness loss and the augmentation regularization loss for regularization.

a) Alignment loss: For unsupervised optical flow estimation tasks, the core metric is the alignment quality. Here we follow the popular brightness constancy assumption that the two frames should have similar pixel intensities so as to utilize the photometric loss $\mathcal{L}_p$ as the main error metric. However, considering aligned pixels do not exist in the internal occluded regions, we only evaluate $\mathcal{L}_p$ on the non-occluded regions and

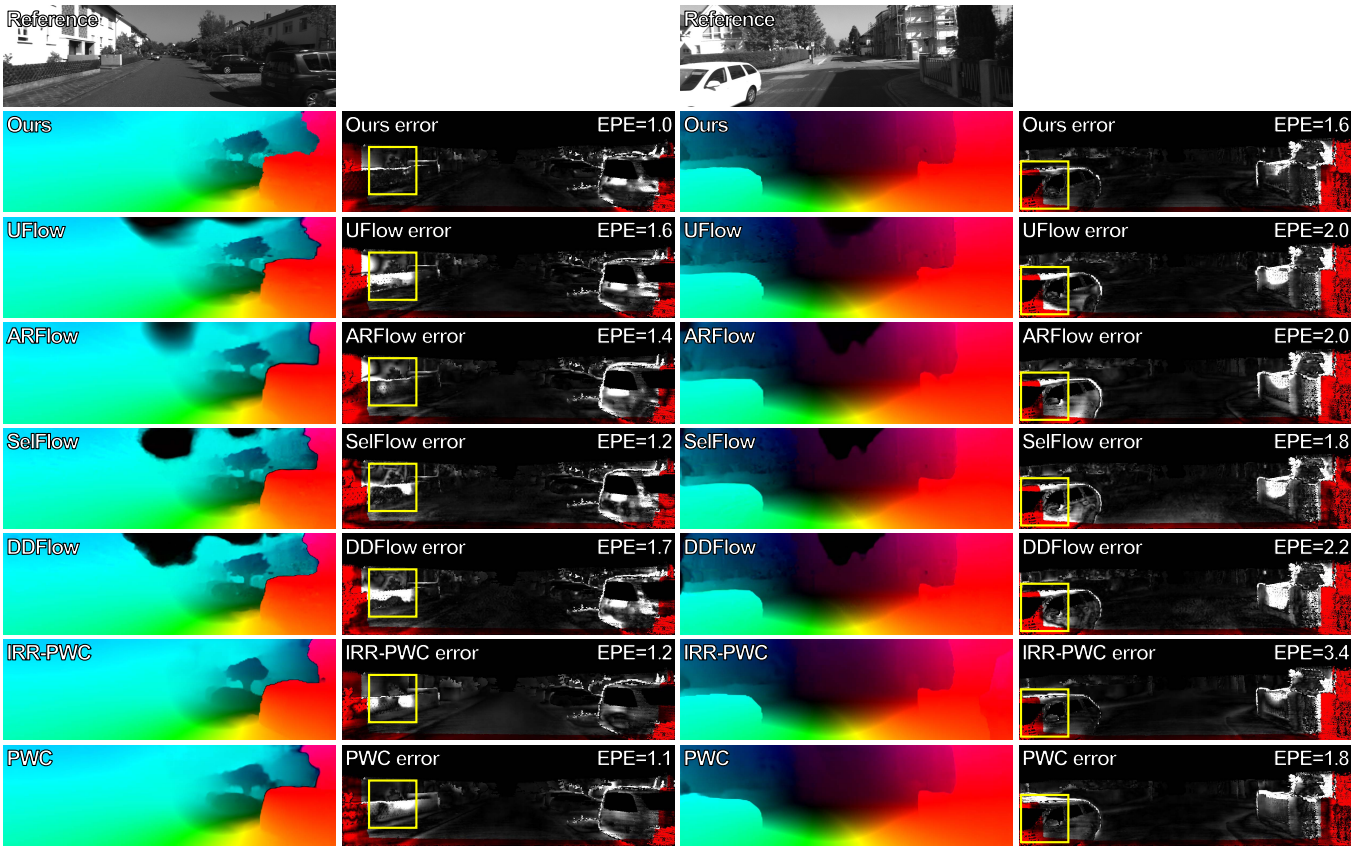


Fig. 6. Comparison on KITTI 2012 benchmark. Results of the unsupervised methods UFlow [59], ARFlow [58], SelFlow [32] and DDFlow [31] and the fully supervised methods PWC-Net [40] and IRR-PWC [35] are shown. For better comparison, error maps are also visualized in the even columns, where areas with larger errors are brighter and the occluded regions are highlighted in red.

the boundary occluded regions. As mentioned before, we apply Boundary Dilated Warping $\mathcal{W}_\Omega$ to each frame instead of the standard warping $\mathcal{W}$. As a result, $\mathcal{L}_p$ is formulated as follows.

$$\mathcal{L}_p = \frac{\sum_{\boldsymbol{p}} \left[\psi(I_1 - \tilde{I}_2) \odot (1 - M_{\Omega,f}) \right]}{\sum_{\boldsymbol{p}} (1 - M_{\Omega,f})} + \frac{\sum_{\boldsymbol{p}} \left[\psi(I_2 - \tilde{I}_1) \odot (1 - M_{\Omega,b}) \right]}{\sum_{\boldsymbol{p}} (1 - M_{\Omega,b})}, \quad (11)$$

where $\psi(x) = (|x| + \epsilon)^q$ is a robust loss with $\epsilon = 0.01$ and $q = 0.4$ in our experiments, and $\odot$ is element-wise multiplier.

To robustly handle the case that brightness constancy assumption is not met, we further apply census transform [46] $\mathcal{C}(\cdot)$ to each of images $I_1, I_2, \tilde{I}_1, \tilde{I}_2$ before the calculation. Census transform is a local binary pattern encoding the local pixel intensity ordering, so that it has a certain capability to overcome the illuminance variation. To substitute $I_1, I_2, \tilde{I}_1, \tilde{I}_2$ by their census transformed versions $\mathcal{C}(I_1), \mathcal{C}(I_2), \mathcal{C}(\tilde{I}_1), \mathcal{C}(\tilde{I}_2)$, we obtain a census loss $\mathcal{L}_c$ of similar to $\mathcal{L}_p$ as in Eq. 11.

Note that computing census loss by the combination of the robust loss function and census transform is first proposed by UnFlow [46]. Here, following UnFlow, we also calculate the census loss based on a robust function and census transform except for two differences. First, the form of our census loss is different from UnFlow. We put the occluded mask in the denominator as normalization, while UnFlow gives a constant penalty to the occlusion pixels to avoid the trivial solution where all pixels being occluded. Second, the robust function is different. In our experiments, we found that the absolute robust function is better than the robust generalized Charbonnier penalty function ($\rho(x) = (x^2 + \epsilon^2)^\gamma$ with $\gamma = 0.45$) of UnFlow.

b) Occlusion inpainting loss: The alignment losses (photometric loss and census loss) defined in Eq. 11 can provide guidance information for optical flow learning in non-occlusion regions. However, as mentioned above, occlusion regions caused by moving objects cannot be aligned essentially, which makes the alignment loss erroneous for optical flow learning in these regions. Without proper guidance information, optical flow estimation is always erroneous in these object occlusion regions. In order to improve optical flow estimation in these object occlusion regions, we propose an occlusion inpainting method based on the contextual relevance clue. We first learn the correspondence between non-occlusion pixels and occlusion pixels using an image inpainting loss, referred to as occlusion inpainting loss for short. This correspondence is represented by the appearance flow estimated by our AppFlowNet-Lite. Then, during the inference process, we use the learned appearance flow to inpaint flow vectors in object occlusion regions based on estimations in non-occlusion

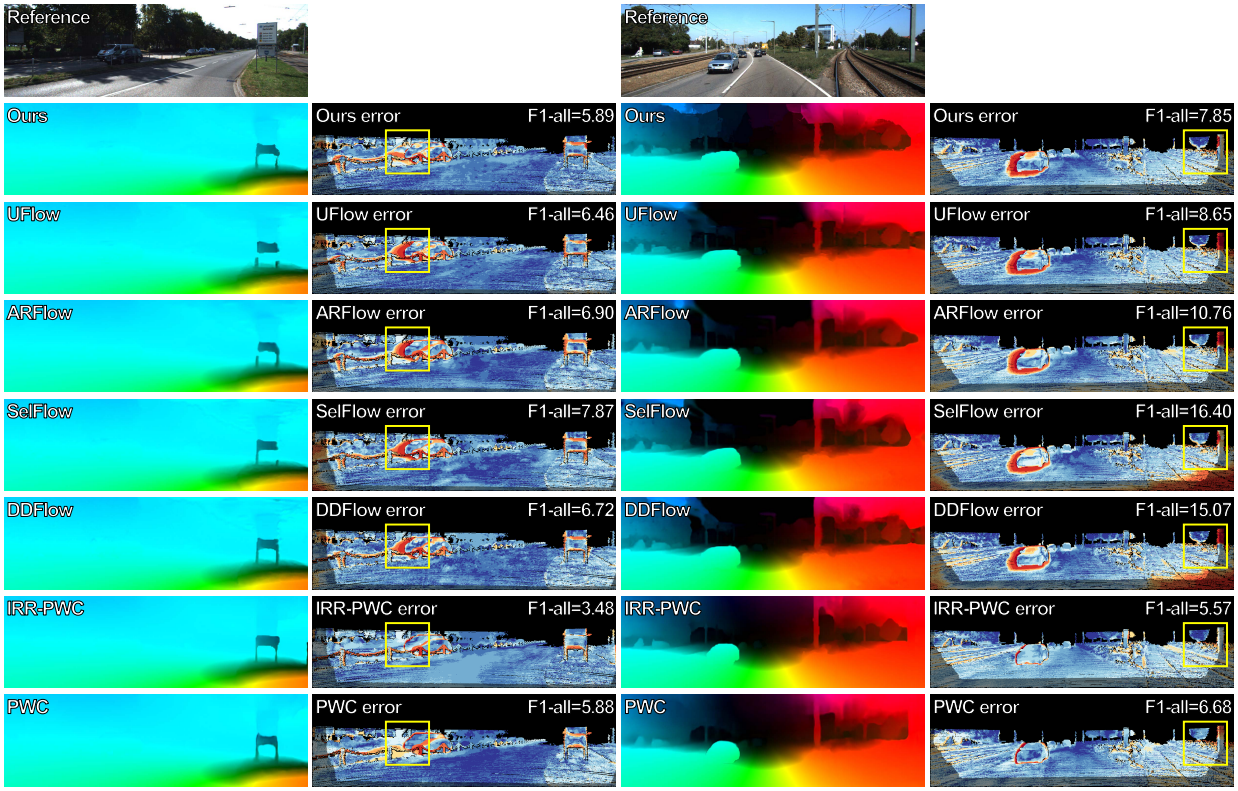


Fig. 7. Comparison on KITTI 2015 benchmark. Error maps are visualized in the even columns, correct pixels are displayed in blue and wrong ones in red.

regions. The occlusion inpainting loss is formulated as follows:

$$\mathcal{L}_{oi} = \sum_{i=0}^{N} \left(\frac{\sum_{\boldsymbol{p}} \left[\psi(I_1^i - \tilde{Q}_1^i) \odot M_{\Omega,f}^i \right]}{\sum_{\boldsymbol{p}} M_{\Omega,f}^i} + \frac{\sum_{\boldsymbol{p}} \left[\psi(I_2^i - \tilde{Q}_2^i) \odot M_{\Omega,b}^i \right]}{\sum_{\boldsymbol{p}} M_{\Omega,b}^i} \right), \quad (12)$$

where $\tilde{Q}_1^i = \mathcal{W}(Q_1^i, A_f^i)$, $\tilde{Q}_2^i = \mathcal{W}(Q_2^i, A_b^i)$, and here $Q_k^i, k \in \{1, 2\}$ are the masked versions of I_k^i by the occlusion maps $M_{\Omega,f}^i, M_{\Omega,b}^i$. That is to say, $Q_k^i(\boldsymbol{p}) = I_k^i(\boldsymbol{p}) \cdot \left(1 - M_{\Omega,\beta}^i(\boldsymbol{p})\right)$, for $\langle k, \beta \rangle \in \{\langle 1, f \rangle, \langle 2, b \rangle\}$. N is the index of the finest scale, being set to 4 in our experiments. Eq. 12 reveals that the reconstruction error between the inpainted image and itself is evaluated on the internal occluded regions only. By minimizing this loss, our AppFlowNet-Lite well learns how to proceed "PatchMatch"-like hole filling for the RGB image, so as to be further applied to the optical flows for refinement. Considering there is no out-of-boundary problem, we just use a standard warping $\mathcal{W}$.

c) Smoothness loss: We also add a regularization term to constrain the smoothness of the optical flows following the idea in [30], which minimizes the first order and second order edge-aware smoothness loss of the predicted optical flows.

$$\mathcal{L}_s = \sum_{\boldsymbol{p}} \sum_{d \in x,y} |\partial_d V_\beta| \exp(-|\partial_d I_k|) + \sum_{\boldsymbol{p}} \sum_{d \in x,y} |\partial_d^2 V_\beta| \exp(-|\partial_d^2 I_k|), \quad (13)$$

where $\langle k, \beta \rangle \in \{\langle 1, f \rangle, \langle 2, b \rangle\}$.

d) Augmentation regularization loss: Augmentation regularization (AR) loss is a self-supervision learning objective that first proposed by ARFlow [58]. Here we follow ARFlow and use the AR loss in our framework to improve the performance. Given the input image pair $\{I_1, I_2\}$, its prediction optical flow V_f and occlusion mask $M_{\Omega,f}$, we first perform transformations to construct the augmented sample:

$$\{\bar{I}_1, \bar{I}_2\}, \overline{V}_f, \overline{M}_{\Omega,f} = \mathcal{T}(\{I_1, I_2\}, V_f, M_{\Omega,f}), \quad (14)$$

where $\mathcal{T}$ is the augmentation transformer, $\{\bar{I}_1, \bar{I}_2\}$ is the image pair after augmentation, $\overline{V}_f$ and $\overline{M}_{\Omega,f}$ are the optical flow and occlusion mask after the same spatial transformation with the image augmentation. Note that the value of the flow vectors in $\overline{V}_f$ are also changed properly to represent the correct motion between the image pair $\{\bar{I}_1, \bar{I}_2\}$. Then, we feed the new image pair into the network and calculate the AR loss by using $\overline{V}_f$ as a pseudo label and using $\overline{M}_{\Omega,f}$ to exclude the unreliable regions in $\overline{V}_f$. The AR loss can be formulated as follows:

$$\mathcal{L}_{ar} = \sum_{\boldsymbol{p}} \left[\psi(\overline{V}_f^* - \overline{V}_f) \odot (1 - \overline{M}_{\Omega,f}) \right], \quad (15)$$

where $\overline{V}_f^*$ is the prediction flow of the image pair $\{\bar{I}_1, \bar{I}_2\}$.

Eventually, our unsupervised loss $\mathcal{L}$ is defined as the combination of the 5 losses above,

$$\mathcal{L} = \lambda_p \mathcal{L}_p + \lambda_c \mathcal{L}_c + \lambda_{oi} \mathcal{L}_{oi} + \lambda_s \mathcal{L}_s + \lambda_{ar} \mathcal{L}_{ar}, \quad (16)$$

where the hyper-parameters are: $\lambda_p = 1$, $\lambda_c = 1$, $\lambda_{oi} = 1$, $\lambda_s = 0.05$ and $\lambda_{ar} = 0.5$.

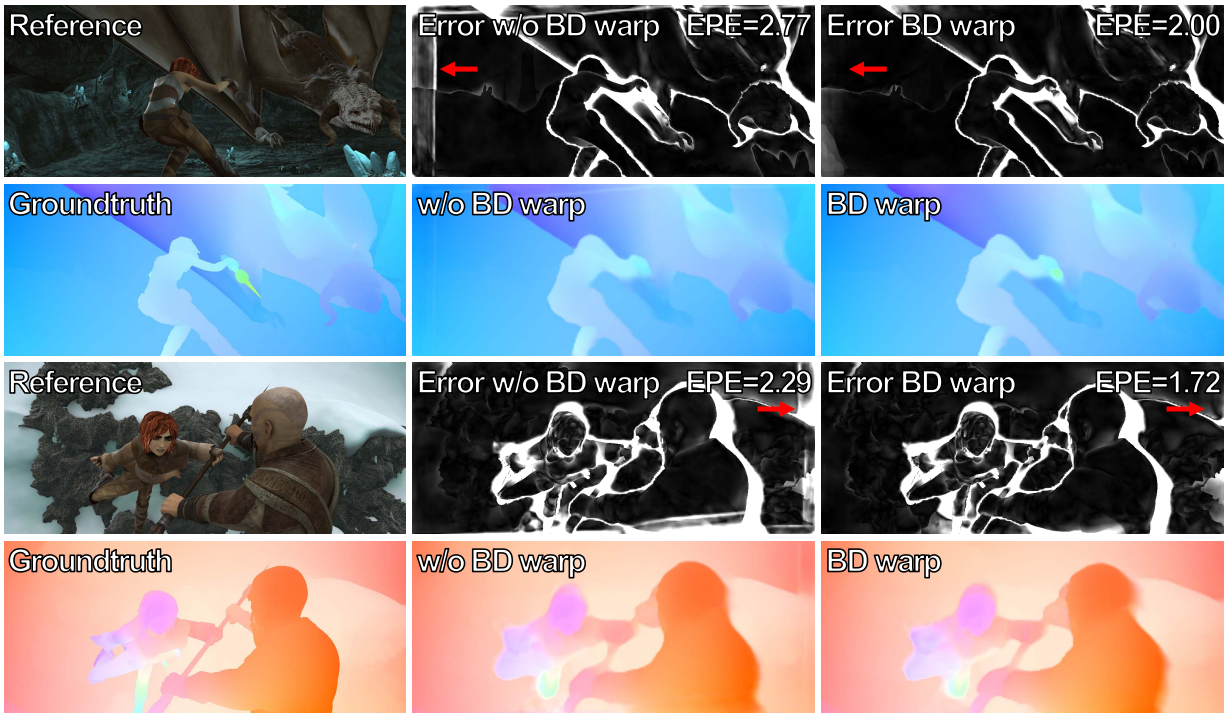
Fig. 8. Comparison on MPI-Sintel Final benchmark, where results of other unsupervised methods UFlow [59], ARFlow [58], SelFlow [32] and DDFlow [31] are shown. Results of fully supervised methods PWC-Net [40] and IRR-PWC [35] are shown in the last two rows. The error maps of the predictions are visualized in the even columns. For better visualization, the zoom-in views of some local patches are depicted in the right bottom corner of each sample.

IV. Experimental Results

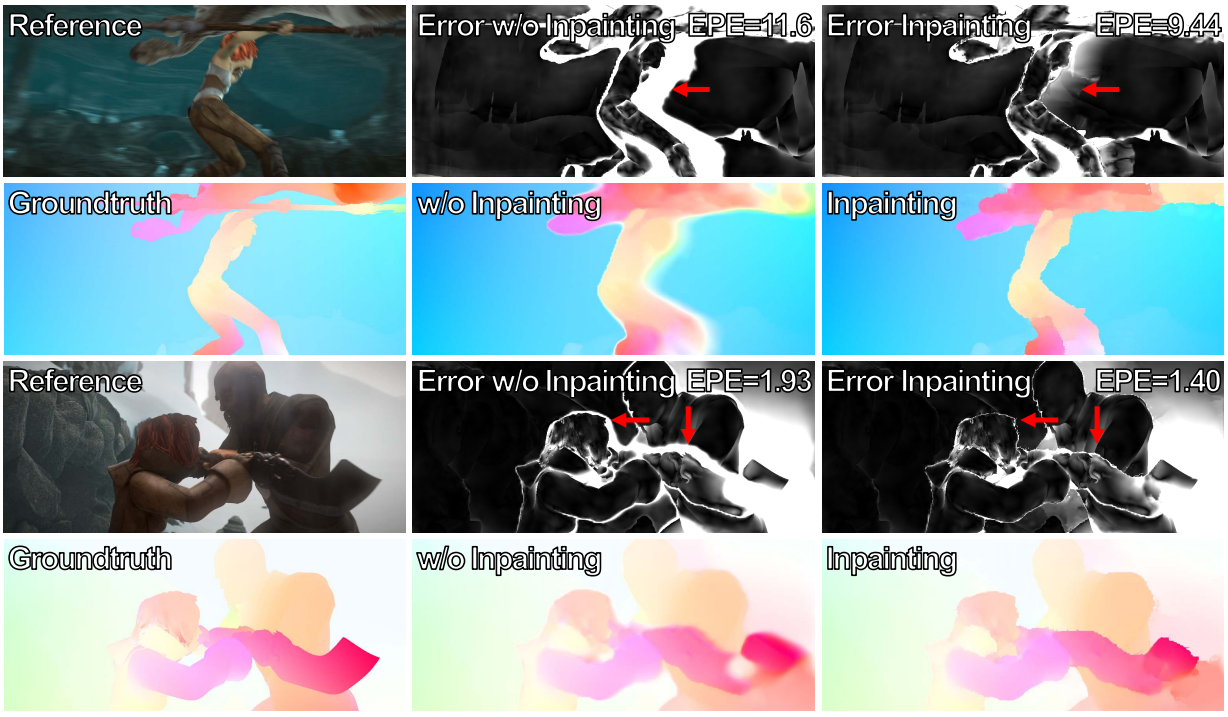
A. Datasets and Implementation Details

We conduct our experiments and comparison on the following 3 datasets. During training, we set different crop sizes for the different datasets, and the crop coordinate $\boldsymbol{p}_0$ is set randomly, but requiring that the border of the cropped image retains at least 8 pixels from the border of the original image.

1) **Flying Chairs.** Flying chairs [16] is a synthetic dataset created by randomly moving chair images, which contains 22,872 pairs for training and 640 pairs for testing. Following DDFlow [31], the training images are used for unsupervised training (no ground truth), and the test images are used for validation. The image size is 384×512 and we set the crop size as 320×320 for our boundary dilated warping during the training process.
2) **KITTI.** KITTI 2012 dataset [25] contains 194 training pairs and 195 test pairs, and KITTI 2015 dataset [5] contains 200 training pairs and 200 test pairs. The two datasets also contain multi-view extension datasets with no ground truth. Following [58], [59], we use the image pairs of the multi-view extension for training, and use the train set of KITTI2012 and KITTI2015 for validation. Specifically, the image size is around 376×1240 and we set the crop size as 320×896 during training. Results are uploaded to KITTI website for benchmark comparison.
3) **MPI-Sintel.** MPI-Sintel [24] is a synthetic dataset. It contains 1,041 training pairs and 564 test pairs. Two different rendering sets are provided, 'Clean' and 'Final'. We use 1,041 pairs from 'Clean' for training. The image size is 436×1024 and we set the crop size as 320×768 during



(a) Comparison of with v.s. without the boundary dilated warping.



(b) Comparison of with v.s. without the occlusion inpainting.

Fig. 9. Visual comparison of with and without the proposed components on MPI-Sintel clean dataset. The error maps are also provided and the obvious difference is highlighted by the red arrows.

training. We also upload results of the test set for online evaluation.

1) Implementation Details: We develop our code with PyTorch, and the whole training was completed in 500k iterations which cost about 48 hours by 2 NVIDIA GeForce GTX 2080Ti GPUs. The total number of parameters of our network is around 5.26M. To stabilize the training, we divide the training process into 3 stages. First, we set the learning rate to 10^{-4} to learn initial optical flows with AppFlowNet-Lite disabled and without occlusion handling. After about 100k iterations,

TABLE II

ABLATION STUDY. WE CONDUCT EXPERIMENTS TO ASSESS THE IMPACT OF DIFFERENT COMPONENTS IN OUR FRAMEWORK. 'OE' MEANS OCCLUSION EXCLUSION WHICH IS TO EXCLUDE THE OCCLUSION REGIONS FROM THE ALIGNMENT LOSSES AS IN UNFLOW [46]. 'AR' MEANS THE AR LOSS PROPOSED BY ARFLOW [58]. 'BD' MEANS THAT THE BOUNDARY DILATED WARPING IS USED TO MODIFY THE ALIGNMENT LOSSES. 'OI' INDICATES THE OCCLUSION INPAINTING METHOD WHERE OUR APPFLOWNET-LITE IS ENABLED AND TRAINED WITH THE OCCLUSION INPAINTING LOSS. THE EPE ERROR OVER ALL PIXELS (ALL), NON-OCCLUDED PIXELS (NOC) AND OCCLUDED PIXELS (OCC) ARE REPORTED FOR DETAIL

OE	AR	BD	OI	Chairs	KITTI 2012			KITTI 2015			Sintel Clean			Sintel Final		
				ALL	ALL	NOC	OCC	ALL	NOC	OCC	ALL	NOC	OCC	ALL	NOC	OCC
				3.89	6.17	1.35	34.70	9.84	2.74	40.56	(4.34)	(2.27)	(20.44)	(5.02)	(2.97)	(20.19)
		✓		3.25	2.10	1.28	6.80	4.31	2.74	11.11	(3.44)	(1.78)	(16.62)	(4.55)	(2.87)	(17.16)
		✓	✓	3.01	2.09	1.28	6.79	4.25	2.72	10.77	(3.25)	(1.84)	(14.49)	(4.12)	(2.68)	(14.81)
✓				3.28	3.86	1.24	18.78	7.96	2.46	31.79	(3.53)	(1.71)	(17.85)	(4.48)	(2.63)	(18.40)
✓		✓		3.05	1.37	0.87	4.29	2.84	2.01	6.41	(2.88)	(1.48)	(14.06)	(4.10)	(2.58)	(15.63)
✓		✓	✓	2.83	1.35	**0.85**	4.27	2.79	1.98	6.27	(2.78)	(1.50)	(12.87)	(3.89)	(2.53)	(13.93)
✓	✓			2.98	1.72	1.06	5.44	3.29	2.20	7.94	(2.94)	(1.51)	(14.63)	(3.85)	(2.42)	(15.20)
✓	✓	✓		2.96	1.34	0.92	3.81	2.61	1.94	6.02	(2.55)	**(1.21)**	(13.46)	(3.57)	(2.21)	(14.31)
✓	✓	✓	✓	**2.53**	**1.33**	0.92	**3.78**	**2.57**	**1.92**	**5.71**	**(2.43)**	(1.24)	**(12.08)**	**(3.35)**	**(2.17)**	**(12.59)**

we exclude the occlusion regions in $\mathcal{L}_p$ to learn accurate optical flows in the non-occluded regions for $100k$ iterations. Finally, our AppFlowNet-Lite is enabled and predictions from the non-occluded regions is propagated to the occluded ones. We use the average endpoint error (EPE) to evaluate the quality of our predicted optical flow. The percentage of erroneous pixels (F1) is also used as an evaluation metric on the test set of KITTI2015.

B. Comparison With Existing Methods

As shown in Table I, we compare our approach with the previous representative works on Flying Chairs, KITTI and Sintel datasets. The proposed approach outperforms all the previous works on all the datasets.

On KITTI2012 online evaluation, we set the state-of-the-art EPE of unsupervised methods to 1.6, which is even better than fully supervised methods such as FlowNet2 [17] and PWC-Net [40]. Moreover, on KITTI2015 online evaluation, we improve the F1-all value from 11.13% of UFlow [59] to 9.81% with 12% improvement, which is also comparable with leading fully supervised methods. On the test benchmark of MPI-Sintel, we achieve 4.26 on the clean and 5.71 on the final, both are better than all the previous unsupervised methods.

We show the visual comparisons with previous leading methods on KITTI benchmark [5], [25] in Fig. 6 and Fig. 7, including unsupervised methods, such as UFlow [59], ARFlow [58], SelFlow [32] and DDFlow [31] and supervised methods such as PWC-Net [40] and IRR-PWC [35]. We label the EPE error on the upper right of the image. Our method produces smaller errors than all the other unsupervised ones. We highlight some regions with yellow boxes in Fig. 6 and Fig. 7. Note that, our method is comparable with or even outperforms the two fully supervised methods [35], [40].

We also compare our method with other unsupervised approaches on Sintel benchmark [24]. Fig. 8 shows two examples as two columns, where the first row shows the two frames and the other rows show the results from ours, UFlow [59], ARFlow [58], SelFlow [32], DDFlow [31], PWC-Net [40] and IRR-PWC [35]. We plot both flows and the error map to demonstrate the differences. As seen from Fig. 8, our method produces sharper predictions in object edges and less noise in other regions. Error from ours is the smallest compared with other unsupervised methods.

C. Ablation Studies

We first conduct ablation studies to discuss the impact of different components in our framework, i.e. the occlusion exclusion (OE) of the alignment losses, the augmentation regularization loss (AR), the boundary dilated warping (BD) and the occlusion inpainting method (OI). Then we compare our occlusion inpainting method with classical image-guided interpolation methods. We report EPE error of all pixels (ALL), non-occluded pixels (NOC) and occluded pixels (OCC) on the validation set of KITTI, Sintel and Flying Chairs datasets. Since our method is designed to improve the estimation of the occlusion region, the OCC-EPE error is dramatically decreased by our method.

1) Boundary Dilated Warping: As shown in Table II, The EPE-OCC error is significantly reduced by using our boundary dilated warping in all different learning settings. In particular, KITTI dataset is collected during driving with camera shakes, resulting in a large number of pixels moving outside the image boundary. The proposed boundary dilated warping improves the EPE-OCC error from 34.70 to 6.80 in KITTI 2012 and from 40.56 to 11.11 in KITTI 2015 when no other occlusion handling method is used. It is because our boundary dilated warping can provide reliable guidance information for the boundary occlusion region, thereby improving the optical flow estimation in these regions.

We further demonstrate the visual comparisons of with and without our boundary dilated warping in Fig. 9(a) on MPI-Sintel Clean dataset. As seen from the results, optical flow in areas close to the image boundaries can be well learned if boundary dilated warping is enabled. The EPE error is

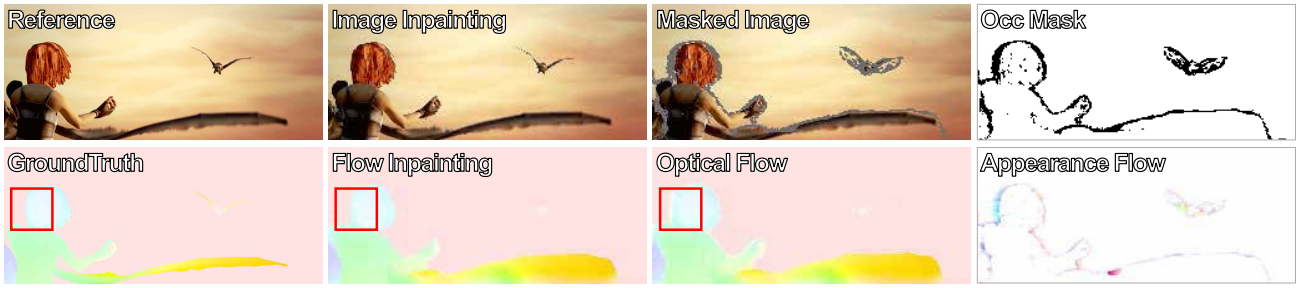


Fig. 10. Illustration of the occlusion inpainting. The first line shows the reference image, image inpainting result, the masked image and the occlusion mask. The second line shows the groundtruth optical flow, optical flow prediction by occlusion inpainting, optical flow before occlusion inpainting and the appearance flow produced by our inpainting network.

TABLE III

QUANTITATIVE COMPARISON OF OCCLUSION FILLING USING OUR OCCLUSION INPAINTING VS. TRADITIONAL INTERPOLATION METHODS. WE REPORT EPE-ALL ON VALIDATION SET OF KITTI AND SINTEL DATASETS

Method	KITTI 2012	KITTI 2015	Sintel Clean	Sintel Final
baseline	1.34	2.61	2.55	3.57
baseline-res	**1.33**	2.60	2.54	3.55
RicFlow [61]	1.49	2.87	2.48	3.51
FGI [62]	1.35	2.69	2.50	3.47
ours	**1.33**	**2.57**	**2.43**	**3.35**

TABLE IV

QUANTITATIVE EVALUATION OF OUR OCCLUSION INPAINTING METHOD IN SUPERVISED SETTING. METHOD MARKED BY '*' MEANS THAT THE MODEL IS PRETRAINED ON SINTEL DATASET USING UNSUPERVISED METHOD

method	Chairs	Sintel Clean			Sintel Final		
	ALL	ALL	NOC	OCC	ALL	NOC	OCC
IRR-PWC [35]	2.08	2.80	–	–	4.13	–	–
baseline	2.09	2.72	1.63	8.50	4.05	2.93	10.27
baseline*	1.98	2.40	1.41	7.89	3.49	2.48	9.19
baseline-res	2.04	2.69	1.62	8.49	3.99	2.88	10.21
baseline-res*	1.96	2.31	1.36	7.63	3.47	2.46	9.13
ours	1.99	2.62	1.56	8.45	3.94	2.82	10.12
ours*	**1.83**	**2.26**	**1.34**	**7.43**	**3.44**	**2.43**	**9.09**

clearly dropped. Please notice the areas highlighted by the red arrow in the error image in Fig. 9(a).

2) Occlusion Inpainting: As shown in Table II, the performance can be improved by our occlusion inpainting method, especially for occlusion regions. Comparing row 2 and row 3, the EPE-OCC error is decreased from 16.62 to 14.49 in Sintel Clean pass and from 17.16 to 14.81 in Sintel Final pass, indicating the effectiveness of our occlusion inpainting method. This is because that our occlusion inpainting method can explicitly introduce image content constrain for optical flow estimation in occlusion regions.

We further show the results of enabling and disabling our occlusion inpainting method on MPI-Sintel clean dataset in Fig. 9(b). As seen, optical flow predictions by our occlusion inpainting method are significantly sharper and more accurate in object edges and occluded regions. We highlight some regions with red arrows in Fig. 9(b) for a clear illustration.

In order to explain what exactly our inpainting network is learning, we also show the occlusion mask, the appearance flow, the image inpainting result and the optical flow inpainting result in Fig. 10. As can be seen, the masked image is successfully restored by a warping operation based on the appearance flow. Moreover, after flow inpainting by using the same appearance flow, the error-prone occlusion regions in optical flow are improved, especially the regions around the head of the human.

3) Occlusion Inpainting vs. Occlusion Interpolation: Table III compares our occlusion inpainting method with traditional image-guided flow interpolation methods, such as RicFlow [60] and Fast Guided global Interpolation (FGI) [61]. The first row is the baseline model trained by using occlusion exclusion, AR loss and our boundary dilated warping. The 'baseline-res' indicates that the appearance flow is used as a residual flow to be added on the predicted flow for updating, which slightly increases the performance. For a fair comparison, We replace the inpainting module in our network architecture with RicFlow and FGI, which means that the same initial optical flow and occlusion mask are used as the input of these interpolation methods compared with our full method. Results show that our learned occlusion inpainting model outperforms traditional interpolation methods especially on KITTI 2015 and Sintel datasets that contain a lot of in-frame occlusion regions caused by dynamic objects.

4) Occlusion Inpainting in Supervised Setting: To further demonstrate the usefulness of our occlusion inpainting method, we conduct a set of experiments in the supervised setting. Following IRR-PWC [35], we train networks on the train set of Flying Chairs dataset and evaluate on the test set of Flying Chairs dataset and the train set of Sintel Clean and Sintel Final. Method marked by '*' means that the model is pre-trained on Sintel dataset using our unsupervised learning method, otherwise, the model is trained from scratch. The results are shown in Table IV. The first line is the performance reported by IRR-PWC [35], whose network structure is the same as our baseline model. From Table IV, we can see that: (1) with unsupervised pretraining on the validation sets, models can converge well and achieve better results, which is also demonstrated by SelFlow [32]; (2) because the guidance information of occlusion region is provided by the ground truth, updating residual flow map in the 'baseline-res' method can improve the performance; (3) the performance can be further improved by our appearance flow warping, which has the contextual propagate ability from non-occlusion regions to occlusion regions.

V. Conclusion

We have presented OIFlow, an occlusion-inpainting optical flow estimation framework for unsupervised learning. It works in a multi-scale manner and is composed of 3 main parts, including a feature extractor to extract the feature maps of the input frames in multiple scales, and a shared initial optical flow estimation block together with an optical flow refinement block working in each scale. Our framework enables the awareness of the occlusion caused by internal objects and the limited image plane. We design a special boundary dilated warping combined with a light version of AppFlowNet to inpaint the occluded regions. We conduct extensive ablation studies to demonstrate the newly-introduced blocks can effectively improve the quality of the predicted optical flows. The results also show that our method significantly outperforms the state-of-the-art unsupervised approach for optical flow estimation.

References

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

[2] N. Bonneel, J. Tompkin, K. Sunkavalli, D. Sun, S. Paris, and H. Pfister, "Blind video temporal consistency," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 196:1–196:9, Nov. 2015.

[3] S. Niklaus and F. Liu, "Softmax splatting for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5437–5446.

[4] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou, "RIFE: Real-time intermediate flow estimation for video frame interpolation," 2020, *arXiv:2011.06294*. [Online]. Available: http://arxiv.org/abs/2011.06294

[5] 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.

[6] 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.

[7] X. Ren, "Local grouping for optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[8] L. Xu, J. Chen, and J. Jia, "A segmentation based variational model for accurate optical flow estimation," in *Proc. ECCV*, 2008, pp. 671–684.

[9] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black, "Optical flow with semantic segmentation and localized layers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3889–3898.

[10] D. Sun, C. Liu, and H. Pfister, "Local layering for joint motion estimation and occlusion detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1098–1105.

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

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

[13] 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.

[14] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. ECCV*, 2004, pp. 25–36.

[15] T. Kroeger, R. Timofte, D. Dai, and L. V. Gool, "Fast optical flow using dense inverse search," in *Proc. ECCV*, 2016, pp. 471–488.

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

[17] 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.

[18] W.-S. Lai, J.-B. Huang, and M.-H. Yang, "Semi-supervised learning for optical flow with generative adversarial networks," in *Proc. NeurIPS*, 2017, pp. 354–364.

[19] Y. Yang and S. Soatto, "Conditional prior networks for optical flow," in *Proc. ECCV*, 2018, pp. 282–298.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] D. Butler, J. Wulff, G. Stanley, and M. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. ECCV*, 2012, pp. 611–625.

[25] 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. (CVPR)*, Jun. 2012, pp. 3354–3361.

[26] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss, "Human-assisted motion annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[27] 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.

[28] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, "Unsupervised deep learning for optical flow estimation," in *Proc. AAAI*, 2017, pp. 1495–1501.

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

[30] 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.

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

[32] 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.

[33] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *Proc. ECCV*, 2016, pp. 286–301.

[34] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li, "StructureFlow: Image inpainting via structure-aware appearance flow," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 181–190.

[35] J. Hur and S. Roth, "Iterative residual refinement for joint optical flow and occlusion estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5747–5756.

[36] 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, pp. 2710–2719.

[37] S. Zhao, Y. Sheng, Y. Dong, E. I.-C. Chang, and Y. Xu, "MaskFlownet: Asymmetric feature matching with learnable occlusion mask," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6278–6287.

[38] J. Xu, R. Ranftl, and V. Koltun, "Accurate optical flow via direct cost volume processing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5807–5815.

[39] G. Yang and D. Ramanan, "Volumetric correspondence networks for optical flow," in *Proc. NeurIPS*, 2019, pp. 794–805.

[40] 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.

[41] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Models matter, so does training: An empirical study of CNNs for optical flow estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1408–1423, Jun. 2020.

[42] T.-W. Hui, X. Tang, and C. C. Loy, "A lightweight optical flow CNN–revisiting data fidelity and regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2555–2569, Aug. 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9018073

[43] J. Wang, Y. Zhong, Y. Dai, K. Zhang, P. Ji, and H. Li, "Displacement-invariant matching cost learning for accurate optical flow estimation," in *Proc. NeurIPS*, 2020, pp. 15220–15231.

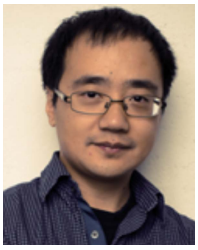
[44] T.-W. Hui and C. C. Loy, "LiteFlowNet3: Resolving correspondence ambiguity for more accurate optical flow estimation," in *Proc. ECCV*, 2020, pp. 169–184.
[45] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proc. ECCV*, 2020, pp. 402–419.
[46] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proc. AAAI*, 2018, pp. 7251–7259.
[47] J. Janai, F. Güey, A. Ranjan, M. Black, and A. Geiger, "Unsupervised learning of multi-frame optical flow with occlusions," in *Proc. ECCV*, 2018, pp. 713–731.
[48] Y. Zhong, P. Ji, J. Wang, Y. Dai, and H. Li, "Unsupervised deep epipolar flow for stationary or dynamic scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12095–12104.
[49] A. Ranjan *et al.*, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12240–12249.
[50] Y. Zou, Z. Luo, and J.-B. Huang, "DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency," in *Proc. ECCV*, 2018, pp. 38–55.
[51] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1983–1992.
[52] L. Liu, G. Zhai, W. Ye, and Y. Liu, "Unsupervised learning of scene flow estimation fusing with local rigidity," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 876–882.
[53] Y. Wang, P. Wang, Z. Yang, C. Luo, Y. Yang, and W. Xu, "UnOS: Unified unsupervised optical-flow and stereo-depth estimation by watching videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8063–8073.
[54] W. Im, T.-K. Kim, and S.-E. Yoon, "Unsupervised learning of optical flow with deep feature similarity," in *Proc. ECCV*, 2020, pp. 172–188.
[55] L. Tian, Z. Tu, D. Zhang, J. Liu, B. Li, and J. Yuan, "Unsupervised learning of optical flow with CNN-based non-local filtering," *IEEE Trans. Image Process.*, vol. 29, pp. 8429–8442, 2020.
[56] Z. Ren, J. Yan, X. Yang, A. Yuille, and H. Zha, "Unsupervised learning of optical flow with patch consistency and occlusion estimation," *Pattern Recognit.*, vol. 103, Jul. 2020, Art. no. 107191.
[57] Z. Ren *et al.*, "STFlow: Self-taught optical flow estimation using pseudo labels," *IEEE Trans. Image Process.*, vol. 29, pp. 9113–9124, 2020.
[58] L. Liu *et al.*, "Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6489–6498.
[59] R. Jonschkowski, A. Stone, J. T. Barron, A. Gordon, K. Konolige, and A. Angelova, "What matters in unsupervised optical flow," in *Proc. ECCV*, 2020, pp. 557–572.
[60] Y. Hu, Y. Li, and R. Song, "Robust interpolation of correspondences for large displacement optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 481–489.
[61] Y. Li, D. Min, M. N. Do, and J. Lu, "Fast guided global interpolation for depth and motion," in *Proc. ECCV*, 2016, pp. 717–733.

Shuaicheng Liu (Member, IEEE) received the B.E. degree from Sichuan University, Chengdu, China, in 2008, and the M.S. and Ph.D. degrees from the National University of Singapore, Singapore, in 2010 and 2014, respectively. Since 2014, he has been an Associate Professor with the School of Information and Communication Engineering, Institute of Image Processing, University of Electronic Science and Technology of China (UESTC). His research interests include computer vision and computer graphics.

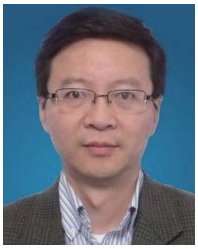
Kunming Luo received the B.E. and M.S. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2016 and 2019, respectively. He is currently a Researcher at Megvii Technology, Chengdu. His research interests include computer vision and deep learning.

Nianjin Ye received the B.E. and M.S. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2017 and 2020, respectively. He is currently a Researcher at Megvii. His research interests include image processing and deep learning.

Chuan Wang (Member, IEEE) received the B.Eng. degree from the University of Science and Technology of China in 2010, and the Ph.D. degree from The University of Hong Kong in 2015. He was a Computer Vision Staff Researcher at Lenovo Group Ltd., Hong Kong. He worked as a Visiting Scholar with the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, China, in 2009, and the State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China, in 2010. In 2018, he started his training program at Megvii. His research interests include video analysis and computer vision.



Jue Wang (Senior Member, IEEE) received the B.E. and M.Sc. degrees from the Department of Automation, Tsinghua University, Beijing, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering from the University of Washington, Seattle, WA, USA, in 2007. He is currently a Senior Director of Megvii. Before that, he has been a Principle Research Scientist at Adobe Research for nine years. His research interests include image and video processing and computational photography. He is a Senior Member of ACM. He received the Microsoft Research Fellowship and the Yang Research Award from the University of Washington in 2006.



Bing Zeng (Fellow, IEEE) received the B.E. and M.Sc. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1983 and 1986, respectively, and the Ph.D. degree in electrical engineering from the Tampere University of Technology, Tampere, Finland, in 1991. From September 1991 to July 1992, he worked as a Postdoctoral Fellow with the University of Toronto. From August 1992 to January 1993, he was a Researcher with Concordia University. He joined The Hong Kong University of Science and Technology (HKUST). After 20 years of service, he returned to UESTC in the Summer of 2013, through Chinas 1000-Talent-Scheme. At UESTC, he leads the Institute of Image Processing to work on image and video processing, 3D and multiview video technology, and visual big data. During his tenure with HKUST and UESTC, he graduated more than 30 master's and Ph.D. students, received about 20 research grants, filed eight international patents, and published more than 250 articles. In 2016, he was elected as an IEEE Fellow for his contributions to image and video coding. He received the Best Associate Editor Award in 2011. In 2014, he was a recipient of the 2nd Class Natural Science Award (the first recipient) from the Ministry of Education of China. He was the General Co-Chair of the IEEE VCIP-2016, held in Chengdu, in November 2016. He serves as the General Co-Chair for PCM-2017. He served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT) for eight years. He is currently on the Editorial Board of *Journal of Visual Communication and Image Representation*.