LiteFlowNet3: Resolving Correspondence Ambiguity for More Accurate Optical Flow Estimation

Tak-Wai Hui[1][0000−0002−1441−9289] and Chen Change Loy[2][0000−0001−5345−1591]

[1] The Chinese University of Hong Kong
[2] Nanyang Technological University
https://github.com/twhui/LiteFlowNet3
twhui@ie.cuhk.edu.hk, ccloy@ntu.edu.sg

Abstract. Deep learning approaches have achieved great success in addressing the problem of optical flow estimation. The keys to success lie in the use of cost volume and coarse-to-fine flow inference. However, the matching problem becomes ill-posed when partially occluded or homogeneous regions exist in images. This causes a cost volume to contain outliers and affects the flow decoding from it. Besides, the coarse-to-fine flow inference demands an accurate flow initialization. Ambiguous correspondence yields erroneous flow fields and affects the flow inferences in subsequent levels. In this paper, we introduce LiteFlowNet3, a deep network consisting of two specialized modules, to address the above challenges. (1) We ameliorate the issue of outliers in the cost volume by amending each cost vector through an adaptive modulation prior to the flow decoding. (2) We further improve the flow accuracy by exploring local flow consistency. To this end, each inaccurate optical flow is replaced with an accurate one from a nearby position through a novel warping of the flow field. LiteFlowNet3 not only achieves promising results on public benchmarks but also has a small model size and a fast runtime.

1 Introduction

Optical flow estimation is a classical problem in computer vision. It is widely used in many applications such as motion tracking, action recognition, video segmentation, 3D reconstruction, and more. With the advancement of deep learning, many research works have attempted to address the problem by using convolutional neural networks (CNNs) [10,11,12,13,18,27,28,31,32]. The majority of the CNNs belongs to the 2-frame method that infers a flow field from an image pair. Particularly, LiteFlowNet [10] and PWC-Net [27] are the first CNNs to propose using the feature warping and cost volume at multiple pyramid levels in a coarse-to-fine estimation. This greatly reduces the number of model parameters from 160M in FlowNet2 [13] to 5.37M in LiteFlowNet and 8.75M in PWC-Net while accurate flow estimation is still maintained.

One of the keys to success for the lightweight optical flow CNNs is the use of cost volume for establishing correspondence at each pyramid level. However, a

cost volume is easily corrupted by ambiguous feature matching [17,26,30]. This causes flow fields that are decoded from the cost volume to become unreliable. The underlying reasons for the existence of ambiguous matching are twofold. First, when given a pair of images, it is impossible for a feature point in the first image to find the corresponding point in the second image, when the latter is occluded. Second, ambiguous correspondence is inevitable in homogeneous regions (*e.g.,* shadows, sky, and walls) of images. Another key to success for the optical flow CNNs is to infer flow fields using a coarse-to-fine framework. However, this approach highly demands an accurate flow initialization from the preceding pyramid level. Once ambiguous correspondence exists, erroneous optical flow is generated and propagates to subsequent levels.

To address the aforementioned challenges, we attempt to make correspondence across images less ambiguous and in turn improves the accuracy of optical flow CNNs by introducing the following specialized CNN modules:

Cost Volume Modulation. Ambiguous feature matching causes outliers to exist in a cost volume. Inaccurate cost vectors need to be amended to allow the correct flow decoding. To deal with occlusions, earlier work improves the matching process by using the offset-centered matching windows [17]. A cost volume is filtered to remove outliers prior to the correspondence decoding [26,30]. However, existing optical flow CNNs [11,12,13,18,28,32,31] infer optical flow from a cost volume using convolutions without explicitly addressing the issue of outliers. We propose to amend each cost vector in the cost volume by using an adaptive affine transformation. A confidence map that pinpoints the locations of unreliable flow is used to facilitate the generation of transformation parameters.

Flow Field Deformation. When the correspondence problem becomes ill-posed, it is very difficult to find correct matching pairs. Local flow consistency and co-occurrence between flow boundaries and intensity edges are commonly used as the clues to regularize flow fields in conventional methods [29,33]. The two principles are also adopted in recent optical flow CNNs [10,11,12]. We propose a novel technique to further improve the flow accuracy by using the clue from local flow consistency. Intuitively, we replace each inaccurate optical flow with an accurate one from a nearby position having similar feature vectors. The replacement is achieved by a meta-warping of the flow field in accordance with a computed displacement field (similar to optical flow but the displacement field no longer represents correspondence). We compute the displacement field by using a confidence-guided decoding from an auto-correlation cost volume.

In this work, we make the first attempt to use cost volume modulation and flow field deformation in optical flow CNNs. We extend our previous work (LiteFlowNet2 [11]) by incorporating the proposed modules for addressing the aforementioned challenges. LiteFlowNet3 achieves promising performance in the 2-frame method. It outperforms VCN-small [31], IRR-PWC [12], PWC-Net+ [28], and LiteFlowNet2 on Sintel and KITTI. Even though SelFlow [18] (a multi-frame method) and HD^3 [32] use extra training data, LiteFlowNet3 outperforms SelFlow on Sintel clean and KITTI while it performs better than HD^3 on Sintel, KITTI 2012, and KITTI 2015 (in foreground region). LiteFlowNet3 does not

suffer from the artifact problem on real-world images as HD^3, while being 7.7 times smaller in model size and 2.2 times faster in runtime.

2 Related Work

Variational Approach. Since the pioneering work of Horn and Schunck [8], the variational approach has been widely studied for optical flow estimation. Brox *et al.* address the problem of illumination change across images by introducing the gradient constancy assumption [3]. Brox *et al.* [3] and Papenberg *et al.* [23] propose the use of image warping in minimizing an energy functional. Bailer *et al.* propose Flow Fields [1], which is a searching-based method. Optical flow is computed by a numerical optimization with multiple propagations and random searches. In EpicFlow [25], Revaud *et al.* use sparse flows as an initialization and then interpolate them to a dense flow field by fitting a local affine model at each pixel based on nearby matches. The affine parameters are computed as the least-square solution of an over-determined system. Unlike EpicFlow, we use an adaptive affine transformation to amend a cost volume. The transformation parameters are implicitly generated in the CNN instead.

Cost Volume Approach. Kang *et al.* address the problem of ambiguous matching by using the offset-centered windows and select a subset of neighboring image frames to perform matching dynamically [17]. Rhemann *et al.* propose to filter a cost volume using an edge-preserving filter [26]. In DCFlow [30], Xu *et al.* exploit regularity in a cost volume and improve the optical flow accuracy by adapting the semi-global matching. With the inspiration of improving cost volume from the above conventional methods, we propose to modulate each cost vector in the cost volume by using an affine transformation prior to the flow decoding. The transformation parameters are adaptively constructed to suit different cost vectors. In particular, DCFlow combines the interpolation in EpicFlow [25] with a complementary scheme to convert a sparse correspondence to a dense one. On the contrary, LiteFlowNet3 applies an affine transformation to all elements in the cost volume but not to sparse correspondence.

Unsupervised and Self-supervised Optical Flow Estimation. To avoid annotating labels, Meister *et al.* propose a framework that uses the difference between synthesized and real images for unsupervised training [21]. Liu *et al.* propose SelFlow that distills reliable flow estimations from non-occluded pixels in a large dataset using self-supervised training [18]. It also uses multiple frames and fine-tunes the self-supervised model in supervised training for improving the flow accuracy further. Unlike the above works, we focus on supervised learning. Even though LiteFlowNet3 is a 2-frame method and trained on a much smaller dataset, it still outperforms SelFlow on Sintel clean and KITTI.

Supervised Learning of Optical Flow. Dosovitskiy *et al.* develop FlowNet [6], the first optical flow CNN. Mayer *et al.* extend FlowNet to estimate disparity and scene flow [20]. In FlowNet2 [13], Ilg *et al.* improve the flow accuracy of FlowNet by cascading several variants of it. However, the model size is increased to over 160M parameters and it also demands a high computation time. Ranjan

et al. develop a compact network SPyNet [24], but the accuracy is not comparable to FlowNet2. Our LiteFlowNet [10], which consists of the cascaded flow inference and flow regularization, has a small model size (5.37M) and comparable performance as FlowNet2. We then develop LiteFlowNet2 for more accurate flow accuracy and faster runtime [11]. LiteFlowNet3 is built upon LiteFlowNet2 with the incorporation of cost volume modulation and flow field deformation for improving the flow accuracy further. A concurrent work to LiteFlowNet is PWC-Net [27], which proposes using the feature warping and cost volume as LiteFlowNet. Sun *et al.* then develop PWC-Net+ by improving the training protocol [28]. Ilg *et al.* extend FlowNet2 to FlowNet3 with the joint learning of occlusion and optical flow [14]. In Devon [19], Lu *et al.* perform feature matching that is governed by an external flow field. On the contrary, our displacement field is used to deform optical flow but not to facilitate feature matching. Hur *et al.* propose IRR-PWC [12], which improves PWC-Net by adopting the flow regularization from LiteFlowNet as well as introducing the occlusion decoder and weight sharing. Yin *et al.* introduce HD^3 for learning a probabilistic pixel correspondence [32], but it requires pre-training on ImageNet. While LiteFlowNet3 learns a flow confidence implicitly but not computed from the probabilistic estimation. Despite HD^3 uses extra training data and 7.7 times more parameters, LiteFlowNet3 outperforms HD^3 on Sintel, KITTI 2012, and KITTI 2015 (in foreground region). LiteFlowNet3 outperforms VCN-small [31] even though the model sizes are similar. Comparing to deformable convolution [5], we perform deformation on flow fields but not on feature maps. Our deformation aims to replace each inaccurate optical flow with an accurate one from a nearby position in the flow field, while deformable convolution aims to augment spatial sampling.

3 LiteFlowNet3

Feature matching becomes ill-posed in homogeneous and partially occluded regions as one-to-multiple correspondence occurs for the first case while one-to-none correspondence occurs for the second case. Duplicate of image structure (so-called "ghosting effect") is inevitable whenever warping is applied to images [15]. The same also applies to feature maps. In coarse-to-fine estimation, erroneous optical flow resulting from the preceding level affects the subsequent flow inferences. To address the above challenges, we develop two specialized CNN modules: *Cost volume Modulation* (CM) and *Flow field Deformation* (FD). We demonstrate the applicability of the modules on LiteFlowNet2 [11]. The resulting network is named as LiteFlowNet3. Figure 1 illustrates a simplified overview of the network architecture. FD is used to refine the previous flow estimate before it is used as a flow initialization in the current pyramid level. In flow inference, the cost volume is amended by CM prior to the flow decoding.

3.1 Preliminaries

We first provide a concise description on the construction of cost volume in optical flow CNNs. Suppose a pair of images I_1 (at time $t = 1$) and I_2 (at time

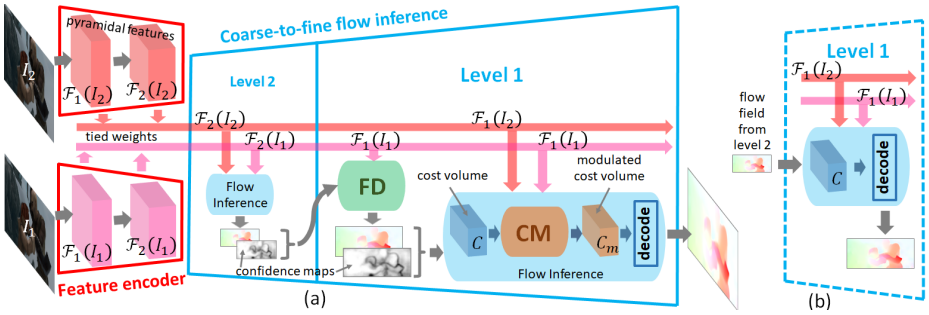


Fig. 1: (a) A simplified overview of LiteFlowNet3. Flow field deformation (FD) and cost volume modulation (CM) together with confidence maps are incorporated into LiteFlowNet3. For the ease of presentation, only a 2-level encoder-decoder structure is shown. The proposed modules are applicable to other levels but not limited to level 1. (b) The optical flow inference in LiteFlowNet2 [11].

$t = 2$) is given. We convert I_1 and I_2 respectively into pyramidal feature maps $\mathcal{F}_1$ and $\mathcal{F}_2$ through a feature encoder. We denote $\mathbf{x}$ as a point in the rectangular domain $\Omega \subset \mathbb{R}^2$. Correspondence between I_1 and I_2 is established by computing the dot product between two high-level feature vectors in the individual feature maps $\mathcal{F}_1$ and $\mathcal{F}_2$ as follows [6]:

$$c(\mathbf{x}; D) = \mathcal{F}_1(\mathbf{x}) \cdot \mathcal{F}_2(\mathbf{x}')/N, \tag{1}$$

where D is the maximum matching radius, $c(\mathbf{x}; D)$ (a 3D column vector with length $2D+1$) is the collection of matching costs between feature vectors $\mathcal{F}_1(\mathbf{x})$ and $\mathcal{F}_2(\mathbf{x}')$ for all possible $\mathbf{x}'$ such that $\|\mathbf{x} - \mathbf{x}'\|_\infty = D$, and N is the length of the feature vector. Cost volume C is constructed by aggregating all $c(\mathbf{x}; D)$ into a 3D grid. Flow decoding is then performed on C using convolutions (or native winner-takes-all approach [17]). The resulting flow field $\mathbf{u} : \Omega \rightarrow \mathbb{R}^2$ provides the dense correspondence from I_1 to I_2. In the following, we will omit variable D that indicates the maximum matching radius for brevity and use $c(\mathbf{x})$ to represent the cost vector at $\mathbf{x}$. When we discuss operations in a pyramid level, the same operations are applicable to other levels.

3.2 Cost Volume Modulation

Given a pair of images, the existence of partial occlusion and homogeneous regions makes the establishment of correspondence very challenging. This situation also occurs on feature space because simply transforming images into feature maps does not resolve the correspondence ambiguity. In this way, a cost volume is corrupted and the subsequent flow decoding is seriously affected. Conventional methods [26,30] address the above problem by filtering a cost volume prior to the decoding. But there has not been any existing works to address this problem for optical flow CNNs. Some studies [2,10,12] have revealed that applying

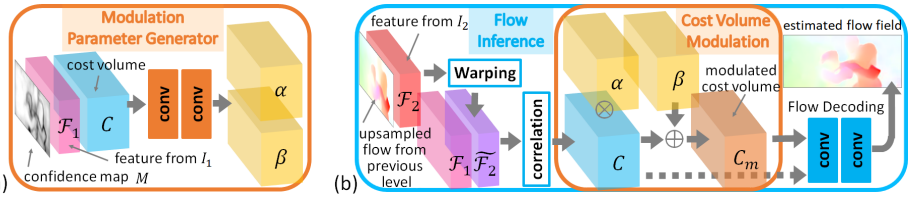


Fig. 2: (a) Modulation tensors (α, β) are adaptively constructed for each cost volume. (b) Cost volume modulation is integrated into the flow inference. Instead of leaving cost volume C unaltered (via the dashed arrow), it is amended to C_m by using the adaptive modulation prior to the flow decoding. Note: "conv" denotes several convolution layers.

feature-driven convolutions on feature space is an effective approach to influence the feed-forward behavior of a network since the filter weights are adaptively constructed. Therefore, we devise to filter outliers in a cost volume by using an adaptive modulation. We will show that our modulation approach is not only effective in improving the flow accuracy but also parameter-efficient.

An overview of cost volume modulation is illustrated in Fig. 2b. At a pyramid level, each cost vector $c(\mathbf{x})$ in cost volume C is adaptively modulated by an affine transformation $(\alpha(\mathbf{x}), \beta(\mathbf{x}))$ as follows:

$$c_m(\mathbf{x}) = \alpha(\mathbf{x}) \otimes c(\mathbf{x}) \oplus \beta(\mathbf{x}), \tag{2}$$

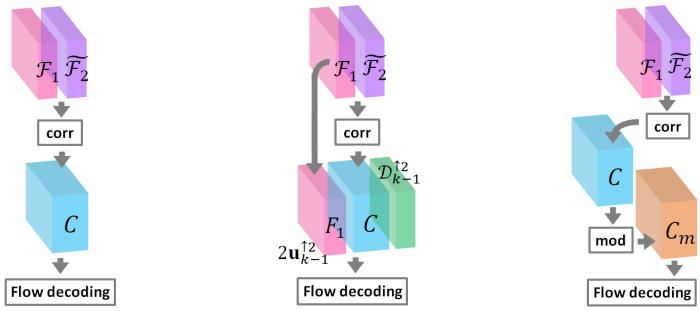
where $c_m(\mathbf{x})$ is the modulated cost vector, "$\otimes$" and "$\oplus$" denote element-wise multiplication and addition, respectively. The dimension of the modulated cost volume is same as the original. This property allows cost volume modulation to be jointly used and trained with an existing network without major changes made to the original network architecture.

To have an efficient computation, the affine parameters $\{\alpha(\mathbf{x}), \beta(\mathbf{x})\}, \forall \mathbf{x} \in \Omega$, are generated altogether in the form of modulation tensors (α, β) having the same dimension as C. As shown in Fig. 2a, we use cost volume C, feature $\mathcal{F}_1$ from the encoder, and confidence map M at the same pyramid level as the inputs to the modulation parameter generator. The confidence map is introduced to facilitate the generation of modulation parameters. Specifically, $M(\mathbf{x})$ pinpoints the probability of having an accurate optical flow at $\mathbf{x}$ in the associated flow field. The confidence map is constructed by introducing an additional output in the preceding optical flow decoder. A sigmoid function is used to constraint its values to $[0, 1]$. We train the confidence map using a L2 loss with the ground-truth label $M_{gt}(\mathbf{x})$ as follows:

$$M_{gt}(\mathbf{x}) = \mathrm{e}^{-\|\mathbf{u}_{gt}(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|^2}, \tag{3}$$

where $\mathbf{u}_{gt}(\mathbf{x})$ is the ground truth of $\mathbf{u}(\mathbf{x})$. An example of predicted confidence maps will be provided in Section 4.2.

Discussion. In the literature, there are two major approaches to infer a flow field from a cost volume as shown in Fig. 3. The first approach (Fig. 3a) is



(a) NO: LiteFlowNet2 [11] (b) FF: PWC-Net+ [28] (c) Ours: LiteFlowNet3

Fig. 3: Augmenting a cost volume under different configurations. (a) NO, (b) FF, and (c) Our solution: Cost volume C is modulated to C_m by using an adaptive affine transformation prior to the flow decoding. Note: "corr" and "mod" denote correlation and modulation, respectively. Correlation is performed on $\mathcal{F}_1$ and warped $\mathcal{F}_2$ (*i.e.*, $\widetilde{\mathcal{F}_2}$).

Table 1: Average end-point error (AEE) and model size of different models trained on FlyingChairs under different augmentations of cost volume.

Augmentations	NO	FF	Ours
Features of I_1	✗	✓	✗
Flow field	✗	✓	✗
Modulation	✗	✗	✓
Number of model parameters (M)	**6.42**	7.16	7.18
Sintel Clean (training set)	2.71	2.70	**2.65**
Sintel Final (training set)	4.14	4.20	**4.02**
KITTI 2012 (training set)	4.20	4.28	**3.95**
KITTI 2015 (training set)	11.12	11.30	**10.65**

to perform flow decoding directly on the cost volume without any augmentation [10,11]. This is similar to the conventional winner-takes-all approach [17] except using convolutions for yielding flow fields rather than argument of the minimum. The second approach (Fig. 3b) feed-forwards the pyramidal features $\mathcal{F}_1$ from the feature encoder [27,28]. It also feed-forwards the upsampled flow field ($2\mathbf{u}^{\uparrow 2}_{k-1}$) and features ($\mathcal{D}^{\uparrow 2}_{k-1}$) from the previous flow decoder (at level $k-1$). Flow decoding is then performed on the concatenation. Our approach (Fig. 3c) is to perform modulation on the cost volume prior to the flow decoding. The effectiveness of the above approaches has not been studied in the literature. Here, we use LiteFlowNet2 [11] as the backbone architecture and train all the models from scratch on FlyingChairs dataset [6]. Table 1 summarizes the results of our evaluation. Even though FF needs 11.5% more model parameters than NO, it attains lower flow accuracy. On the contrary, our modulation approach that has just 0.28% more parameters than FF outperforms the compared methods on all

Fig. 4: Replacing an inaccurate optical flow $\mathbf{u}(\mathbf{x}_1)$ with an accurate optical flow $\mathbf{u}(\mathbf{x}_2)$ through a meta-warping governed by displacement $\mathbf{d}(\mathbf{x}_1)$.

the benchmarks, especially KITTI 2012 and KITTI 2015. This indicates that a large CNN model does not always perform better than a smaller one.

3.3 Flow Field Deformation

In coarse-to-fine flow estimation, a flow estimate from the preceding decoder is used as a flow initialization for the subsequent decoder. This highly demands the previous estimate to be accurate. Otherwise, erroneous optical flow is propagated to subsequent levels and affects the flow inference. Using cost volume modulation alone is not able to address this problem. We explore local flow consistency [29,33] and propose to use a meta-warping for improving the flow accuracy.

Intuitively, we refine a given flow field by replacing each inaccurate optical flow with an accurate one from a nearby position using the principle of local flow consistency. As shown in Fig. 4, suppose an optical flow $\mathbf{u}(\mathbf{x}_1)$ is inaccurate. With some prior knowledge, 1) a nearby optical flow $\mathbf{u}(\mathbf{x}_2)$ such that $\mathbf{x}_2 = \mathbf{x}_1 + \mathbf{d}(\mathbf{x}_1)$ is known to be accurate as indicated by a confidence map; 2) the pyramidal features of I_1 at $\mathbf{x}_1$ and $\mathbf{x}_2$ are similar *i.e.,* $\mathcal{F}_1(\mathbf{x}_1) \sim \mathcal{F}_1(\mathbf{x}_2)$ as indicated by an auto-correlation cost volume. Since image points that have similar feature vectors have similar optical flow in a neighborhood, we replace $\mathbf{u}(\mathbf{x}_1)$ with a clone of $\mathbf{u}(\mathbf{x}_2)$.

The previous analysis is just for a single flow vector. To cover the whole flow field, we need to find a displacement vector for every position in the flow field. In other words, we need to have a displacement field for guiding the meta-warping of flow field. We use a warping mechanism that is similar to image [13] and feature warpings [10,27]. The differences are that our meta-warping is limited to two channels and the physical meaning of the introduced displacement field no longer represents correspondence across images.

An overview of flow field deformation is illustrated in Fig. 5b. At a pyramid level, we replace $\mathbf{u}(\mathbf{x})$ with an neighboring optical flow by warping of $\mathbf{u}(\mathbf{x})$ in accordance to the computed displacement $\mathbf{d}(\mathbf{x})$ as follows:

$$\mathbf{u}_d(\mathbf{x}) = \mathbf{u}\left(\mathbf{x} + \mathbf{d}(\mathbf{x})\right). \tag{4}$$

In particular, not every optical flow needs an amendment. Suppose $\mathbf{u}(\mathbf{x_0})$ is very accurate, then no flow warping is required *i.e.,* $\mathbf{d}(\mathbf{x_0}) \sim \mathbf{0}$.

To generate the displacement field, the location of image point having similar feature as the targeted image point needs to be found. This is accomplished by

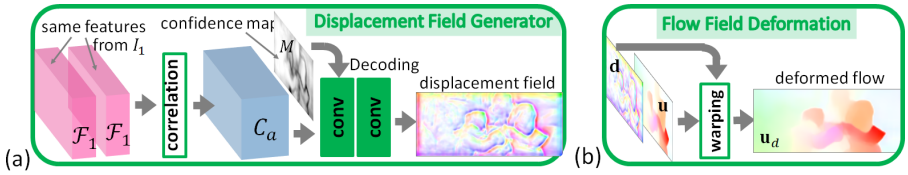


Fig. 5: (a) Displacement field $\mathbf{d}$ is constructed according to auto-correlation cost volume C_a and confidence map M. (b) Flow field $\mathbf{u}$ is warped to $\mathbf{u}_d$ in accordance to $\mathbf{d}$. Flow deformation is performed before $\mathbf{u}$ is used as an initialization for the flow inference. Note: "conv" denotes several convolution layers.

decoding from an auto-correlation cost volume. The procedure is similar to flow decoding from a normal cost volume [6]. As shown in Fig. 5a, we first measure the feature similarity of targeted point at $\mathbf{x}$ and its surrounding points at $\mathbf{x}'$ by computing auto-correlation cost vector $c_a(\mathbf{x}; D)$ between features $\mathcal{F}_1(\mathbf{x})$ and $\mathcal{F}_1(\mathbf{x}')$ as follows:

$$c_a(\mathbf{x}; D) = \mathcal{F}_1(\mathbf{x}) \cdot \mathcal{F}_1(\mathbf{x}')/N, \tag{5}$$

where D is the maximum matching radius, $\mathbf{x}$ and $\mathbf{x}'$ are constrained by $\|\mathbf{x} - \mathbf{x}'\|_\infty = D$, and N is the length of the feature vector. The above equation is identical to Eq. (1) except using features from I_1 only. Auto-correlation cost volume C_a is then built by aggregating all cost vectors into a 3D grid.

To avoid trivial solution, confidence map M associated with flow field $\mathbf{u}$ that is constructed by the preceding flow decoder (same as the one presented in Section 3.2) is used to guide the decoding of displacement from C_a. As shown in Fig. 5a, we use cost volume C_a for the auto-correlation of $\mathcal{F}_1$ and confidence map M at the same pyramid level as the inputs to the displacement field generator. Rather than flow decoding from the cost volume as the normal descriptor matching [6], our displacement decoding is performed on the auto-correlation cost volume and is guided by the confidence map.

4 Experiments

Network Details. LiteFlowNet3 is built upon LiteFlowNet2 [11]. Flow inference is performed from levels 6 to 3 (and 2) with the given image resolution as level 1. Flow field deformation is applied prior to the cascaded flow inference while cost volume modulation is applied in the descriptor matching unit. We do not apply the modules to level 6 as no significant improvement on flow accuracy can be observed (and level 2 due to large computational load). Each module uses four 3×3 convolution layers followed by a leaky rectified linear unit except to use a 5×5 filter in the last layer at levels 4 and 3. Confidence of flow prediction is implicitly generated by introducing an additional convolution layer in a flow decoder. Weight sharing is used on the flow decoders and proposed modules. This variant is denoted by the suffix "S".

Training Details. For a fair comparison, we use the same training sets as other optical flow CNNs in the literature [6,10,11,12,13,19,24,27,28,31]. We use the same training protocol (including data augmentation and batch size) as LiteFlowNet2 [11]. We first train LiteFlowNet2 on FlyingChairs dataset [6] using the stage-wise training procedure [11]. We then integrate brand new modules, cost volume deformation and flow field modulation, into LiteFlowNet2 to form LiteFlowNet3. The newly introduced CNN modules are trained with a learning rate of 1e-4 while the other components are trained with a reduced learning rate of 2e-5 for 300K iterations. We then fine-tune the whole network on FlyingThings3D [20] with a learning rate 5e-6 for 500K iterations. Finally, we fine-tune LiteFlowNet3 respectively on a mixture of Sintel [4] and KITTI [22], and KITTI training sets with a learning rate 5e-5 for 600K iterations. The two models are also re-trained with reduced learning rates and iterations same as LiteFlowNet2.

4.1 Results

We evaluate LiteFlowNet3 on the popular optical flow benchmarks including Sintel clean and final passes [4], KITTI 2012 [7], and KITTI 2015 [22]. We report average end-point error (AEE) for all the benchmarks unless otherwise explicitly specified. More results are available in the supplementary material [9].
Preliminary Discussion. The majority of optical flow CNNs including LiteFlowNet3 are 2-frame methods and use the same datasets for training. However, HD^3 [32] is pre-trained on ImageNet ($>$ 10M images). SelFlow [18] uses Sintel movie ($\sim$ 10K images) and multi-view extensions of KITTI ($>$ 20K images) for self-supervised training. SENSE [16] uses SceneFlow dataset [20] ($>$ 39K images) for pre-training. While SelFlow also uses more than two frames to boost the flow accuracy. Therefore, their evaluations are not directly comparable to the majority of the optical flow CNNs in the literature.
Quantitative Results. Table 2 summarizes the AEE results of LiteFlowNet3 and the state-of-the-art methods on the public benchmarks. With the exception of HD^3 [32], SelFlow [18], and SENSE [16], all the compared CNN models are trained on the same datasets and are the 2-frame method. Thanks to the cost volume modulation and flow field deformation, LiteFlowNet3 outperforms these CNN models including the recent state-of-the-art methods IRR-PWC [12] and VCN-small [31] on both Sintel and KITTI benchmarks. Despite the recent state-of-the-art methods HD^3 and SelFlow (a multi-frame method) use extra training data, LiteFlowNet3 outperforms HD^3 on Sintel, KITTI 2012, and KITTI 2015 (Fl-fg). Our model also performs better than SelFlow on Sintel clean and KITTI. It should be noted that LiteFlowNet3 has a smaller model size and a faster runtime than HD^3 and VCN [31] (a larger variant of VCN-small). We also perform evaluation by dividing AEE into matched and unmatched regions (error over regions that are visible in adjacent frames or only in one of two adjacent frames, respectively). As revealed in Table 3, LiteFlowNet3 achieves the best results on both matched and unmatched regions. Particularly, there is a large improvement on unmatched regions comparing to LiteFlowNet2. This indicates that the proposed modules are effective in addressing correspondence ambiguity.

Table 2: AEE results on the public benchmarks. (Notes: The values in parentheses are the results of the networks on the data they were trained on, and hence are not directly comparable to the others. The best in each category is in bold and the second best is underlined. For KITTI 2012, "All" (or "Noc") represents the average end-point error in total (or non-occluded areas). For KITTI 2015, "Fl-all" (or "-fg") represents the percentage of outliers averaged over all (or foreground) pixels. Inliers are defined as end-point error < 3 pixels or 5%. †Using additional training sets. ‡A multi-frame method.)

Method	Sintel Clean		Sintel Final		KITTI 2012			KITTI 2015			
	train	test	train	test	train	test (All)	test (Noc)	train	train (Fl-all)	test (Fl-fg)	test (Fl-all)
FlowNetS [6]	(3.66)	6.96	(4.44)	7.76	7.52	9.1	-	-	-	-	-
FlowNetC [6]	(3.78)	6.85	(5.28)	8.51	8.79	-	-	-	-	-	-
FlowNet2 [13]	(1.45)	4.16	(2.19)	5.74	(1.43)	1.8	1.0	(2.36)	(8.88%)	8.75%	11.48%
FlowNet3 [14]	(1.47)	4.35	(2.12)	5.67	(1.19)	-	-	(1.79)	-	-	8.60%
SPyNet [24]	(3.17)	6.64	(4.32)	8.36	3.36	4.1	2.0	-	-	43.62%	35.07%
Devon [19]	-	4.34	-	6.35	-	2.6	1.3	-	-	19.49%	14.31%
PWC-Net [27]	(2.02)	4.39	(2.08)	5.04	(1.45)	1.7	0.9	(2.16)	(9.80%)	9.31%	9.60%
PWC-Net+ [28]	(1.71)	3.45	(2.34)	4.60	(0.99)	<u>1.4</u>	<u>0.8</u>	(1.47)	(7.59%)	7.88%	7.72%
IRR-PWC [12]	(1.92)	3.84	(2.51)	4.58	-	1.6	0.9	(1.63)	(5.32%)	<u>7.52%</u>	7.65%
SENSE [16]†	(1.54)	3.60	(2.05)	4.86	(1.18)	1.5	-	(2.05)	(9.69%)	9.33%	8.16%
HD^3 [32]†	(1.70)	4.79	(1.17)	4.67	(0.81)	1.4	**0.7**	(1.31)	(4.10%)	9.02%	**6.55%**
SelFlow [18]†,‡	(1.68)	3.75	(1.77)	**4.26**	(0.76)	1.5	0.9	(1.18)	-	12.48%	8.42%
VCN-small [31]	(1.84)	3.26	(2.44)	4.73	-	-	-	(1.41)	(5.5%)	-	7.74%
LiteFlowNet [10]	(1.35)	4.54	(1.78)	5.38	(1.05)	1.6	<u>0.8</u>	(1.62)	(5.58%)	7.99%	9.38%
LiteFlowNet2 [11]	(1.30)	3.48	(1.62)	4.69	(0.95)	<u>1.4</u>	**0.7**	(1.33)	(4.32%)	7.64%	7.62%
LiteFlowNet3	(1.32)	**2.99**	(1.76)	<u>4.45</u>	(0.91)	**1.3**	**0.7**	(1.26)	(3.82%)	7.75%	7.34%
LiteFlowNet3-S	(1.43)	<u>3.03</u>	(1.90)	4.53	(0.94)	**1.3**	**0.7**	(1.39)	(4.35%)	**6.96%**	<u>7.22%</u>

Qualitative Results. Examples of optical flow predictions on Sintel and KITTI are shown in Figs. 6 and 7, respectively. AEE evaluated on the respective training sets is also provided. For Sintel, the flow fields resulting from LiteFlowNet3 contain less artifacts when comparing with the other state-of-the-art methods. As shown in the second row of Fig. 7, a portion of optical flow over the road fence cannot be recovered by LiteFlowNet2 [11]. On the contrary, it is fully recovered by HD^3 [32] and LiteFlowNet3. Flow bleeding is observed over the road signs for LiteFlowNet2 as illustrated in the third and fourth rows of Fig. 7 while HD^3 and LiteFlowNet3 do not have such a problem. Despite HD^3 is pre-trained on ImageNet and uses 7.7 times more model parameters than LiteFlowNet3, there are serious artifacts on the generated flow fields as shown in the second column of Fig. 7. The above observations suggest that LiteFlowNet3 incorporating the cost volume modulation and flow field deformation is effective in generating optical flow with high accuracy and less artifacts.

Runtime and Model Size. We measure runtime using a Sintel image pair (1024×436) on a machine equipped with Intel Xeon E5 2.2GHz and NVIDIA GTX 1080. Timing is averaged over 100 runs. LiteFlowNet3 needs 59ms for computation and has 5.2M parameters. When weight sharing is not used, the model size is 7.5M. The runtimes of the state-of-the-art 2-frame methods HD^3 [32] and IRR-PWC [12] are 128ms and 180ms, respectively. While HD^3 and IRR-PWC have 39.9M and 6.4M parameters, respectively.

Table 3: AEE results on the testing sets of Sintel. (Note: †Using additional training sets.)

Models	All		Matched		Unmatched	
	Clean	Final	Clean	Final	Clean	Final
FlowNet2 [13]	4.16	5.74	1.56	2.75	25.40	30.11
Devon [19]	4.34	6.35	1.74	3.23	25.58	31.78
PWC-Net+ [28]	3.45	4.60	1.41	2.25	20.12	23.70
IRR-PWC [12]	3.84	4.58	1.47	2.15	23.22	24.36
SENSE [16]†	3.60	4.86	1.38	2.30	21.75	25.73
HD^3 [32]†	4.79	4.67	1.62	2.17	30.63	24.99
LiteFlowNet2 [11]	3.48	4.69	1.33	2.25	20.64	24.57
LiteFlowNet3	**2.99**	**4.45**	**1.15**	**2.09**	**18.08**	**23.68**

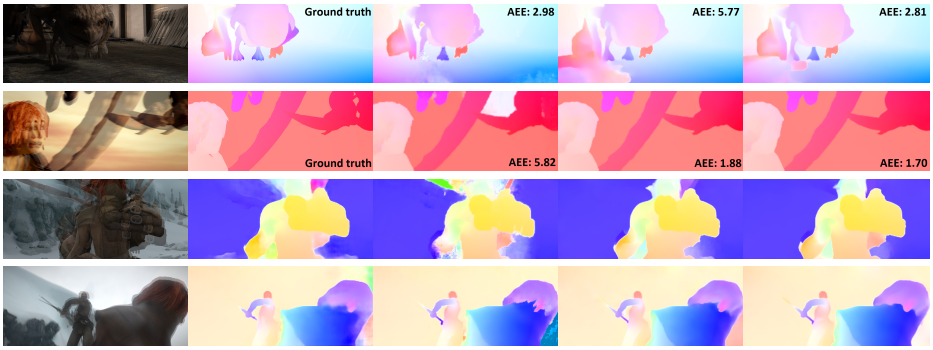


Fig. 6: Examples of flow fields on Sintel training set (Clean pass: first row, Final pass: second row) and testing set (Clean pass: third row, Final pass: forth row).

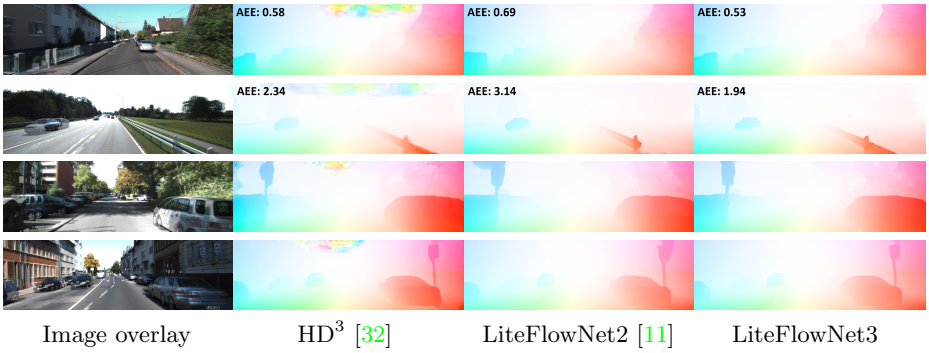


Fig. 7: Examples of flow fields on KITTI training set (2012: first row, 2015: second row) and testing set (2012: third row, 2015: fourth row).

Table 4: AEE results of variants of LiteFlowNet3 having some of the components disabled. (Note: The symbol "-" indicates that confidence map is not being used.)

Settings	NO	CM-	CMFD-	CM	CMFD
Cost Volume **M**odulation	✗	✓	✓	✓	✓
Flow Field **D**eformation	✗	✗	✓	✗	✓
Confidence map	✗	✗	✗	✓	✓
Sintel clean (training set)	2.78	2.66	2.63	2.65	**2.59**
Sintel final (training set)	4.14	4.09	4.06	4.02	**3.91**
KITTI 2012 (training set)	4.11	4.02	4.06	3.95	**3.88**
KITTI 2015 (training set)	11.31	11.01	10.97	10.65	**10.40**

Fig. 8: Examples of flow fields on Sintel Final (top two rows) and KITTI 2015 (bottom two rows) generated by different variants of LiteFlowNet3. Note: NO = No proposed modules are used, CM = **C**ost Volume **M**odulation, CMFD = **C**ost Volume **M**odulation and **F**low Field **D**eformation, and the suffix "-" indicates that confidence map is not being used.

4.2 Ablation Study

To study the role of each proposed component in LiteFlowNet3, we disable some of the components and train the resulting variants on FlyingChairs. The evaluation results on the public benchmarks are summarized in Table 4 and examples of flow fields are illustrated in Fig. 8.

Cost Volume Modulation and Flow Deformation. As revealed in Table 4, when only cost volume modulation (CM) is incorporated to LiteFlowNet3, it performs better than its counterpart (NO) neither using modulation nor deformation on all the benchmarks, especially KITTI 2015. When both of cost volume modulation and flow field formation (CMFD) are utilized, it outperforms the others and achieves in a large improvement on KITTI 2015. Examples of visual performance are demonstrated in Fig. 8. For Sintel, we can observe a large discrepancy in flow color of the human arm between NO and ground

(a) Confidence map (b) Original flow field (c) Displacement (d) Deformed flow

Fig. 9: An example of flow field deformation. The darker a pixel in the confidence map, the more chance the associated optical flow is not correct.

truth. On the contrary, flow color is close to ground truth when CM and CMFD are enabled. Particularly, the green artifact is successfully removed in CMFD. In the example of KITTI, the car's windshield and triangle road sign in NO are not completely filled with correct optical flow. In comparison with CM, the missed flow can be recovered using CMFD only. This indicates that flow field deformation is more efficient in "hole filling" than cost volume modulation.

Confidence Map. Variants CM and CMFD, as revealed in Table 4, perform better than their counterparts CM- and CMFD- with confidence map disabled. For the example of Sintel in Fig. 8, the green artifact is greatly reduced when comparing CM- with CM. Optical flow of the human arm is partially disappeared in CMFD-, while it is recovered in CMFD. The corresponding confidence map is illustrated in Fig. 9a. It indicates that optical flow near the human arm is highly unreliable. Similar phenomenon can also be observed in the example of KITTI. Through pinpointing the flow correctness, the use of confidence map facilitates both cost volume modulation and flow field deformation.

Displacement Field. As shown in Fig. 9c, the active region of the displacement field (having strong color intensity) is well-coincided with the active region of the confidence map (having strong darkness, so indicating high probability of being incorrect flow) in Fig. 9a. The deformed flow field in Fig. 9d has not only less artifacts but also sharper motion boundaries and a lower AEE when comparing to the flow field without meta-warping in Fig. 9b.

5 Conclusion

Correspondence ambiguity is a common problem in optical flow estimation. Ambiguous feature matching causes outliers to exist in a cost volume and in turn affects the decoding of flow from it. Besides, erroneous optical flow can be propagated to subsequent pyramid levels. We propose to amend the cost volume prior to the flow decoding. This is accomplished by modulating each cost vector through an adaptive affine transformation. We further improve the flow accuracy by replacing each inaccurate optical flow with an accurate one from a nearby position through a meta-warping governed by a displacement field. We also propose to use a confidence map to facilitate the generation of modulation parameters and displacement field. LiteFlowNet3, which incorporates the cost volume modulation and flow field deformation, not only demonstrates promising performance on public benchmarks but also has a small model size and a fast runtime.

References

1. Bailer, C., Taetz, B., Stricker, D.: Flow Fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. ICCV pp. 4015–4023 (2015)
2. Brabandere, B.D., Jia, X., Tuytelaars, T., Gool, L.V.: Dynamic filter networks. NIPS (2016)
3. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. ECCV pp. 25–36 (2004)
4. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. ECCV pp. 611–625 (2012)
5. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. ICCV pp. 764–773 (2017)
6. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: FlowNet: Learning optical flow with convolutional networks. ICCV pp. 2758–2766 (2015)
7. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? CVPR pp. 3354–3361 (2012)
8. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Arifical Intelligence **17**, 185–203 (1981)
9. Hui, T.W., Loy, C.C.: Supplementary material for LiteFlowNet3: Resolving Correspondence Ambiguity for More Accurate Optical Flow Estimation (2020)
10. Hui, T.W., Tang, X., Loy, C.C.: LiteFlowNet: A lightweight convolutional neural network for optical flow estimation. CVPR pp. 8981–8989 (2018)
11. Hui, T.W., Tang, X., Loy, C.C.: A lightweight optical flow CNN – Revisiting data fidelity and regularization. TPAMI (2020). https://doi.org/10.1109/TPAMI.2020.2976928
12. Hur, J., Roth, S.: Iterative residual refinement for joint optical flow and occlusion estimation. CVPR pp. 5754–5763 (2019)
13. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet2.0: Evolution of optical flow estimation with deep networks. CVPR pp. 2462–2470 (2017)
14. Ilg, E., Saikia, T., Keuper, M., Brox, T.: Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. ECCV pp. 626–643 (2018)
15. Janai, J., Güney, F., Ranjan, A., Black, M., Geiger, A.: Unsupervised learning of multi-frame optical flow with occlusions. ECCV pp. 713–731 (2018)
16. Jiang, H., Sun, D., Jampani, V., Lv, Z., Learned-Miller, E., Kautz, J.: SENSE: a shared encoder network for scene-flow estimation. ICCV pp. 3195–3204 (2019)
17. Kang, S.B., Szeliski, R., Chai, J.: Handling occlusions in dense multi-view stereo. CVPR pp. 103–110 (2001)
18. Liu, P., Lyu, M., King, I., Xu, J.: SelFlow: Self-supervised learning of optical flow. CVPR pp. 4566–4575 (2019)
19. Lu, Y., Valmadre, J., Wang, H., Kannala, J., Harandi, M., Torr, P.H.S.: Devon: Deformable volume network for learning optical flow. WAVC pp. 2705–2713 (2020)
20. Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. CVPR pp. 4040–4048 (2016)
21. Meister, S., Hur, J., Roth, S.: UnFlow: Unsupervised learning of opticalflow with a bidirectional census loss. AAAI pp. 7251–7259 (2018)

22. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. CVPR pp. 3061–3070 (2015)
23. Papenberg, N., Bruhn, A., Brox, T., Didas, S., Weickert, J.: Highly accurate optic flow computation with theoretically justified warping. IJCV **67**(2), 141–158 (2006)
24. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. CVPR pp. 4161–4170 (2017)
25. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: EpicFlow: Edge-preserving interpolation of correspondences for optical flow. CVPR pp. 1164–1172 (2015)
26. Rhemann, C., Hosni, A., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual correspondence and beyond. CVPR pp. 3017–3024 (2011)
27. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. CVPR pp. 8934–8943 (2018)
28. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Models matter, so does training: An empirical study of CNNs for optical flow estimation. TPAMI (2019). https://doi.org/10.1109/TPAMI.2019.2894353
29. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L^1 optical flow. BMVC (2009)
30. Xu, J., Ranftl, R., Koltun, V.: Accurate optical flow via direct cost volume processings. CVPR pp. 1289–1297 (2017)
31. Yang, G., Ramanan, D.: Volumetric correspondence networks for optical flow. NeurIPS (2019)
32. Yin, Z., Darrell, T., Yu, F.: Hierarchical discrete distribution decomposition for match density estimation. CVPR pp. 6044–6053 (2019)
33. Zimmer, H., Bruhn, A., Weickert, J.: Optic flow in harmony. IJCV **93**(3), 368–388 (2011)