Optical Flow in Dense Foggy Scenes using Semi-Supervised Learning

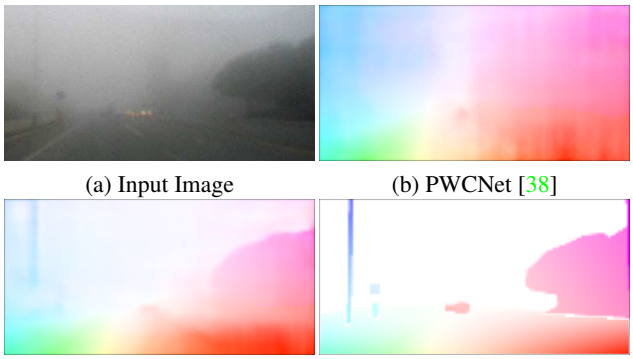
Wending Yan*[1], Aashish Sharma*[1], and Robby T. Tan[1,2]
[1]National University of Singapore, [2]Yale-NUS College
eleyanw@nus.edu.sg, aashish.sharma@u.nus.edu, robby.tan@{nus,yale-nus}.edu.sg

Abstract

In dense foggy scenes, existing optical flow methods are erroneous. This is due to the degradation caused by dense fog particles that break the optical flow basic assumptions such as brightness and gradient constancy. To address the problem, we introduce a semi-supervised deep learning technique that employs real fog images without optical flow ground-truths in the training process. Our network integrates the domain transformation and optical flow networks in one framework. Initially, given a pair of synthetic fog images, its corresponding clean images and optical flow ground-truths, in one training batch we train our network in a supervised manner. Subsequently, given a pair of real fog images and a pair of clean images that are not corresponding to each other (unpaired), in the next training batch, we train our network in an unsupervised manner. We then alternate the training of synthetic and real data iteratively. We use real data without ground-truths, since to have ground-truths in such conditions is intractable, and also to avoid the overfitting problem of synthetic data training, where the knowledge learned on synthetic data cannot be generalized to real data testing. Together with the network architecture design, we propose a new training strategy that combines supervised synthetic-data training and unsupervised real-data training. Experimental results show that our method is effective and outperforms the state-of-the-art methods in estimating optical flow in dense foggy scenes.

1. Introduction

Fog is a common and inevitable weather phenomenon. It degrades visibility by weakening the background scene information, and washing out the colors of the scene. This degradation breaks the Brightness Constancy Constraint (BCC) and Gradient Constancy Constraint (GCC) used in existing optical flow methods. To our knowledge, none of the existing methods can handle dense foggy scenes robustly. This is because most of them (e.g. [42, 6, 14, 38, 30, 43]) are designed under the assumption of clear visibility.



(a) Input Image (b) PWCNet [38]
(c) **Our Result** (d) Ground-Truth

Figure 1: (a) Input dense foggy image (first frame). (b) Optical flow result from the existing baseline method PWCNet [38]. We can observe that the result is erroneous and the method cannot handle dense fog. As shown in (c), compared to it, our method performs more robustly.

One of the possible solutions is to render synthetic fog images based on the commonly used physics model (i.e., the Koschmieder model [18]), and then to train a network on the synthetic fog images and their corresponding optical flow ground-truths in a supervised manner. While in our investigation, it works to some extent, when applied to real dense fog images in the testing stage, it does not perform adequately. The main cause is the domain gap between the synthetic and real fog images. The synthetic images are too crude to represent the complexity of real fog images. This problem can be fixed by using real fog images, instead of synthetic fog images for training. However, to obtain the correct optical flow ground-truths for real fog images is extremely challenging [3].

Another possible solution is to defog the real fog images using an existing defogging method (e.g., [39, 7, 12, 1, 45, 21]), and then to estimate the flow using an existing optical flow method. This two-stage solution, however, is not effective either. First, existing defogging methods are not designed for optical flow, hence their outputs might not be optimum for flow computation. Second, defogging, particularly for dense fog, is still an open problem. Hence, the

*Both authors contributed equally to this work.
†This work is supported by MOE2019-T2-1-130.

outputs of existing defogging methods are still inadequate to make the estimation of optical flow accurate and robust.

Our goal in this paper is to estimate the optical flow from dense fog images robustly. To achieve the goal, we introduce a novel deep learning method that integrates domain transformation (e.g. [15, 47]) and optical flow estimation in a single framework. Initially, given a pair of synthetic fog images, its corresponding clean images and optical flow ground-truths, in one training batch, we train our network in a supervised manner. Subsequently, given a pair of real fog images and a pair of clean images that are not corresponding to each other (unpaired data), in the next training batch, we train our network in an unsupervised manner. The training of synthetic and real data are carried out alternately. We use the synthetic data with ground-truths to guide the network to learn the transformation correctly, so that we can mitigate the fake-content generation problem that is commonly observed in unpaired training [47, 25, 36]. We use the real data without ground-truths to avoid the overfitting problem of the synthetic data training, where the knowledge learned from synthetic data in the training cannot be generalized to real data in the testing.

In essence, our method is a semi-supervised method. Our domain transformation enables our network to learn directly from real data without ground-truths. When the training input is a pair of clean images, our domain transformation renders the corresponding foggy images, and our optical flow module estimates the flow map. Moreover, from the rendered foggy images, our optical flow module also estimates the flow map. Hence, these two different flow maps must be identical. If they are not, then we can back-propagate the error. The same mechanism applies when the training input is a pair of foggy images. Another advantage of our architecture is that the transformation and optical flow modules can benefit each other: Our domain transformation helps our optical flow module reconstruct the fog-invariant cost volume, and our optical flow module enables our domain transformation module to distinguish some objects from the background through the flow information. As a summary, here are our contributions:

- We introduce an architecture that integrates domain transformation and optical flow modules in one framework. The two modules work in mutual cooperation benefiting each other at the feature pyramid levels.
- We propose a training strategy combining synthetic data with ground-truths, clean and fog real data without ground-truths in one integrated learning process.
- We provide a domain adaptive method, which can predict optical flow from both clean and fog images. We also show the effectiveness of using photometric and hazeline [2] constraints to make our network learn better about optical flow and fog.

2. Related Work

Many methods have been proposed to tackle optical flow estimation ([8] for a comprehensive survey). More recently, deep learning is widely used in optical flow methods. Dosovitskiy et. al. [6] design FlowNetS and FlowNetC based on the U-Net architecture [34]. Their method is a pioneer work in showing the possibility of using a deep-learning method to solve the optical flow problem. Ilg et al. [14] design FlowNet2 by stacking multiple FlowNetS and FlowNetC networks. FlowNet2 is trained in a stack-wise manner, and thus is not end-to-end. Sun et al. [38] propose PWCNet. Its performance is comparable to FlowNet2, yet it is significantly smaller in terms of network parameters. All these methods are fully supervised and trained using synthetic data. In contrast, our method uses semi-supervised learning, employing labeled synthetic and unlabeled real data.

Jason et al. [16] propose an unsupervised learning method for flow estimation, for the first time. Ren et al. [33] publish a method with a more complex structure. These two methods simply use the brightness constancy and motion smoothness losses. Some other methods combined depth, ego-motion and optical flow together, such as Yin and Shi [44] and Ranjan et al. [31]. These methods, however, use three independent networks to estimate depth, ego-motion and optical flow, and require camera calibration. Generally, the performance of the current unsupervised methods cannot be as accurate and sharp as that of the fully supervised methods. To take the advantages of both fully supervised and unsupervised learning, Lai et al. [19] proposed a semi-supervised method, which uses the discriminative loss from the warping difference between two frames. Recently, there is progress in unsupervised optical flow (e.g. [46, 41, 31]), under the assumption that the input images are clean. Liu at al. [26] propose a self-supervised method for learning optical flow from unlabeled data. They use photometric loss to obtain reliable flow estimations, which are later used as ground-truths for training. To our knowledge, none of these methods are designed to handle dense foggy scenes. While some previous non-learning-based works (e.g. [28]) can handle illumination variations in the images, these methods also cannot handle dense foggy scenes. This is because fog is more than just intensity/illumination changes in the images, and robustness to illumination variations does not necessarily ensure robustness to fog.

Some works address the problem of semantic segmentation under fog (e.g. [35, 5]). However, they employ a gradual learning scheme, where the network is first trained on labeled synthetic fog data. Then, the network is used to generate flow results on light real fog data. The network is then trained again on labeled synthetic fog data and light fog real data, for which the results predicted before are used as ground-truths. The entire process is repeated for dense

fog real data. While this learning scheme is simple to implement, it has a few problems. First, it makes the entire learning scheme manual. In contrast, our method is completely end-to-end trainable and requires no manual intervention. Second, the results predicted for real data in the previous stage are used as ground-truths for training in the next stage, which could be erroneous. This can lead the network to learning inaccurate flow estimations. In contrast, our method uses accurate flow ground truths from synthetic data, to learn on rendered real fog data. This ensures that the flow network always learns from correct flow ground-truths.

One possible solution of estimating optical flow in foggy scenes is a two-stage solution: defog first and optical flow estimation afterwards. Many methods in defogging/dehazing have been proposed. (see [23] for a comprehensive review). A few methods are based on deep learning, e.g. [4, 32, 20, 21]. All these methods are based on a single image, and thus can cause inconsistent defogging outputs, which in turn causes the violation of the BCC and GCC. Moreover, defogging, particularly for dense fog is still an open problem. Hence, the outputs of existing defogging methods can still be inadequate to generate robust optical flow estimation.

3. Proposed Method

3.1. Network Architecture

Optical Flow Network Our optical flow module consists of two encoders E_f, E_c and a decoder D_{of}, which are shown in Fig. 2, where subscripts f, c, and of stand for fog, clean, and optical flow, respectively. The two encoders (E_f and E_c,) extract features from the fog and clean input images respectively. They have the same architecture, but independent weights. As recent works [30, 38] show that pyramid features improve the estimation of optical flow, we design our encoders in the same way. Our decoder correlates the pyramid features from two input images to form a cost volume, which is used to predict optical flow. Since our decoder receives features from the two encoders working on different domains (fog and clean), it encourages the two encoders to generate domain adaptive features. This domain adaptation ensures that robust optical flow is generated from the two domain inputs.

Domain Transformation Network Our domain transformation module is formed by the encoders, E_f and E_c, and two decoders, D_f and D_c. The fog encoder, E_f, takes the fog images as the input, and outputs feature pyramids. The clean decoder, D_c, processes the features, and constructs the clean version of the input images. The other encoder, E_c, does the same, however instead of fog images, it takes clean images as the input. The fog decoder, D_f, processes the features produced by E_c, and transforms them to fog images. To ensure the proper quality of our transformed clean and fog images, we employ the discriminative loss [11]. While domain transformation is not our main goal, the quality of the transformed images can affect the optical flow result. Note that, we employ feature pyramids in computing the features, so that the same features can also be used by our optical flow network.

3.2. Semi-Supervised Training Strategy

To train our network, ideally we should use real fog data with the corresponding optical flow ground-truths. Unfortunately, to obtain the ground-truths of real fog images is extremely intractable. The best possible technology we can employ currently is LIDAR sensors. However, LIDAR captures only sparse depths and stationary objects. Moreover, it has limited depth range and its accuracy is affected by fog dense particles [3]. An alternative solution is to use synthetic fog images, whose corresponding optical flow is easy to obtain. However, it is known that there are significant gaps between synthetic and real fog images. Synthetic fog images are too simplistic and cannot represent real fog and its complexity in many conditions. Because of these problems, we utilize real clean (no fog) images to help our network learn about fog, clean background scenes, and optical flow. While there are domain gaps between clean real images and fog real images, we bridge the gaps through our domain transformation network.

Our training strategy includes datasets both with and without ground-truths, involving real fog images, synthetic fog images, and real clean images. The reason we use the synthetic fog images is because, they can help guide the network to transform the features from different image domains more correctly by mitigating the generation of fake contents during the transformation. The whole process of our training strategy can be separated into three stages: Synthetic-fog training stage, real-clean training stage, and real-fog training stage.

3.3. Synthetic-Data Training Stage

Given synthetic fog images, their corresponding synthetic clean background images, and their corresponding optical flow ground-truths, we can train our network in a fully supervised manner. First, to train the optical flow module: $\{E_f, E_c, D_{of}\}$, we use EPE (End-Point Error) losses between the predicted optical flow and the corresponding ground-truths for both synthetic fog and clean input images:

$$\mathcal{L}_{\text{EPE}_s^f}(E_f, D_{of}) = \mathbb{E}_{(x_{s1}^f, x_{s2}^f)}\left[\|\widehat{of}^{\,f} - of_{gt}^f\|_2\right], \quad (1)$$

$$\mathcal{L}_{\text{EPE}_s^c}(E_c, D_{of}) = \mathbb{E}_{(x_{s1}^c, x_{s2}^c)}\left[\|\widehat{of}^{\,c} - of_{gt}^c\|_2\right], \quad (2)$$

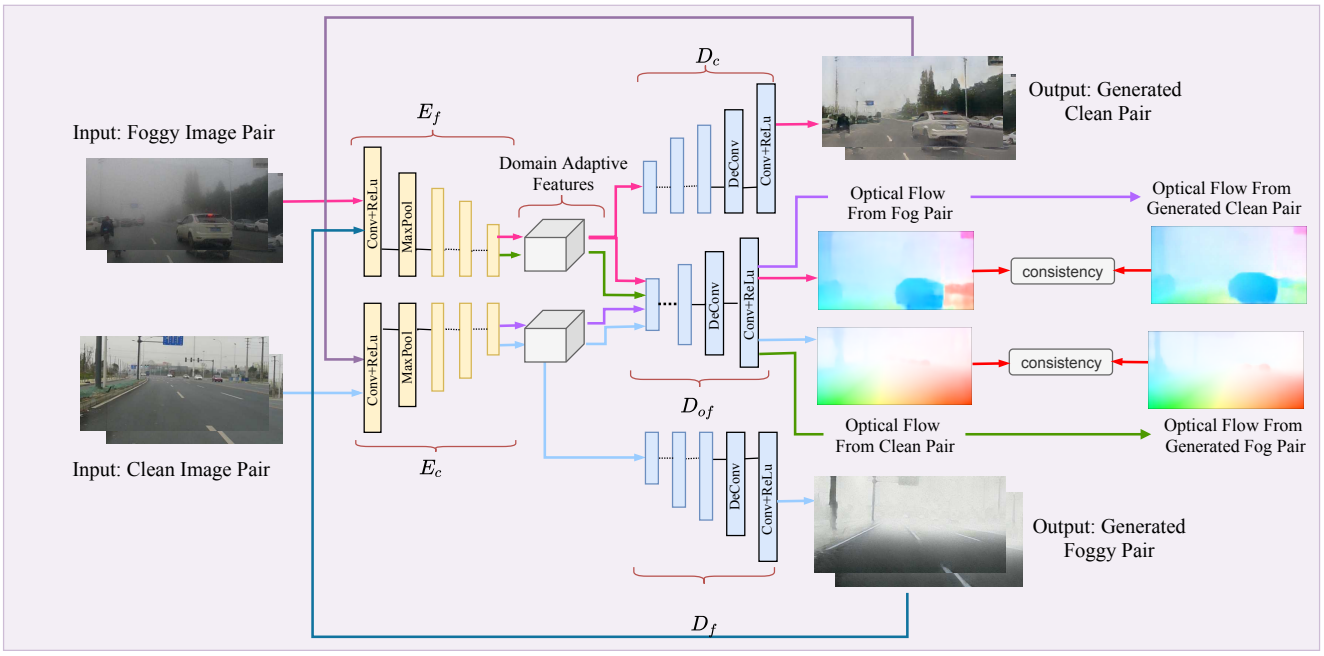


Figure 2: Overall architecture of our network.

with:

$$\widehat{of}^{f} = D_{of}[(E_f[x^f_{s1}], E_f[x^f_{s2}])], \quad (3)$$

$$\widehat{of}^{c} = D_{of}[(E_c[x^c_{s1}], E_c[x^c_{s2}])], \quad (4)$$

where (x^f_{s1}, x^f_{s2}) and (x^c_{s1}, x^c_{s2}) are the synthetic fog and synthetic clean image pairs. of^f_{gt} and of^c_{gt} are the optical flow ground-truths of the synthetic fog and synthetic clean images respectively.

To train the domain transformation module: $\{E_f, D_c\}$ and $\{E_c, D_f\}$, we define L1 losses:

$$\mathcal{L}_{L1^f_s}(E_f, D_c) = \mathbb{E}_{x^f_s}\left[\|\hat{x}^c_s - x^c_{gt}\|_1\right], \quad (5)$$

$$\mathcal{L}_{L1^c_s}(E_c, D_f) = \mathbb{E}_{x^c_s}\left[\|\hat{x}^f_s - x^f_{gt}\|_1\right], \quad (6)$$

where, $\hat{x}^c_s = D_c[(E_f[x^f_s])]$, and $\hat{x}^f_s = D_f[(E_c[x^c_s])]$ are the rendered clean and fog images, respectively. x^c_{gt}, x^f_{gt} are the synthetic clean and synthetic fog ground-truth images, respectively. In addition, we also apply the discriminative loss [11] to ensure that the transformations from clean-to-fog images and from fog-to-clean images are consistent with the appearance of synthetic fog and synthetic clean images.

3.4. Real Clean Data Training Stage

In this stage, we use the real clean images without optical flow ground-truths and without real fog image ground-truths to train the network. As shown in the second row of Fig. 2, first, we compute the optical flow directly from the input real clean images, x^c_{r1}, x^c_{r2}:

$$\widehat{of}^{c} = D_{of}[(E_c[x^c_{r1}], E_c[x^c_{r2}]). \quad (7)$$

Concurrently, we transform the input clean images, x^c_{r1}, x^c_{r2} to fog images, $\hat{x}^f_{r1}, \hat{x}^f_{r2}$:

$$\hat{x}^f_{r1} = D_f[E_c[x^c_{r1}]], \quad (8)$$

$$\hat{x}^f_{r2} = D_f[E_c[x^c_{r2}]]. \quad (9)$$

From the rendered fog images, $\hat{x}^f_{r1}, \hat{x}^f_{r2}$, subsequently we transform them further to obtain the rendered clean images, $\hat{\hat{x}}^c_{r1}, \hat{\hat{x}}^c_{r2}$:

$$\hat{\hat{x}}^c_{r1} = D_c[E_f[\hat{x}^f_{r1}]], \quad (10)$$

$$\hat{\hat{x}}^c_{r2} = D_c[E_f[\hat{x}^f_{r2}]]. \quad (11)$$

At the same time, we also compute the optical flow from the rendered fog images, $\hat{x}^f_{r1}, \hat{x}^f_{r2}$:

$$\widehat{\widehat{of}}^{f} = D_{of}[\hat{x}^f_{r1}, \hat{x}^f_{r2}]. \quad (12)$$

The whole process above, from the input real clean images, x^c_{r1}, x^c_{r2} to the rendered clean, $\hat{\hat{x}}^c_{r1}, \hat{\hat{x}}^c_{r2}$, and to the estimated optical flow, $\widehat{of}^{c}$ and $\widehat{\widehat{of}}^{f}$ is a feedforward process. Initially, we rely on the network's weights learned from synthetic data for this feedforward process. To refine the weights, we train the network further using our current real data. The training is based on a few losses: Transformation consistency, EPE, discriminative, and hazeline losses.

Transformation Consistency Loss To train the domain transformation modules: E_f, E_c, D_f, D_c, we define our consistency loss between the clean input images, x^c_{r1}, x^c_{r2},

and the rendered clean images, $\hat{\hat{x}}^c_{r1}, \hat{\hat{x}}^c_{r2}$, as:

$$\mathcal{L}_{\text{CON}^c_r}(E_f, E_c, D_f, D_c) = \mathbb{E}_{x^c_r}\big[\|x^c_{r1} - \hat{\hat{x}}^c_{r1}\|_1 + \|x^c_{r2} - \hat{\hat{x}}^c_{r2}\|_1\big]. \quad (13)$$

This loss is a pixel-wise computation, since the real clean and rendered clean images must share the same optical flow up to the pixel level. In this backpropagation process, we keep D_{of} frozen.

EPE Loss Since we do not have the optical flow ground-truths of the real clean input images, to train our modules E_f and D_{of}, we define the EPE loss by comparing the predicted optical flow from the real clean input images and the predicted optical flow from the rendered fog images:

$$\mathcal{L}_{\text{EPE}^c_r}(E_f, D_{of}) = \mathbb{E}_{(x^c_{r1}, x^c_{r2})}\big[\|\widehat{of}^{\,c}, \widehat{\widehat{of}^{\,f}}\|_2\big], \quad (14)$$

where $\widehat{of}^{\,c}, \widehat{\widehat{of}^{\,f}}$ are the predicted optical flow fields from the input clean images, and from the rendered fog images, respectively. During the backpropagation of this EPE loss, only E_f and D_{of} are updated, and the rest remain frozen.

Discriminative Loss To train the transformation modules, E_c, D_f, we use the discriminative loss [11] to ensure that the rendered fog images look as real as possible (since we do not have the corresponding real-fog ground-truths). For this purpose, we define our discriminative loss as:

$$\mathcal{L}_{\text{GAN}^c_r}(E_c, D_f) = \mathbb{E}_{x^c_r}\big[(\log(1 - Dis_f[D_f[(E_c[x^c_r]]])\big], \quad (15)$$

where $Dis[.]$ is our discriminative module, which assesses the outputs of D_f. We keep other modules frozen, while updating the weights of E_c, D_f.

Hazeline Loss Since we do not have the ground-truths of the corresponding real fog images, applying the discriminative loss alone will be insufficient to train the modules E_c, D_f properly. Improper training can cause the generation of fake contents [47, 25, 36]. The guidance of the synthetic training data (Sec. 3.3) can mitigate the problem; since synthetic fog images are rendered using a physics model, and thus E_c, D_f learn the underlying physics model from the synthetic fog images. To strengthen the transformation even further, we add a loss based on the following physics model [12, 39, 1] (also used in the rendering of our synthetic fog images):

$$x^f(\mathbf{x}) = x^c(\mathbf{x})\alpha(\mathbf{x}) + (1 - \alpha(\mathbf{x}))\mathbf{A}, \quad (16)$$

where x^f is the fog image, x^c is the clean (no fog) image ground-truth. $\mathbf{A}$ is the atmospheric light. α is the attenuation factor, and $\mathbf{x}$ is the pixel location.

Berman et al. [1] observe that in the RGB space, x^f, x^c, and $\mathbf{A}$ are colinear, due to the linear combination described in the model (Eq. (16)). Unlike Berman et al.'s method, instead of using the RGB space, we use the 2D chromaticity space [39]; since, there is no robust way to estimate the intensity of the atmospheric light [37]. The chromaticity of the clean input image is defined as:

$$\gamma^c_{r,ch} = \frac{x^c_{r,ch}}{x^c_{r,R} + x^c_{r,G} + x^c_{r,B}}, \quad (17)$$

where the index $ch = \{R, G, B\}$ is the RGB color channel. Accordingly, the chromaticity of the rendered fog image by E_c, D_f is defined as:

$$\sigma^c_{r,ch} = \frac{\hat{x}^f_{r,ch}}{\hat{x}^f_{r,R} + \hat{x}^f_{r,G} + \hat{x}^f_{r,B}}. \quad (18)$$

Lastly, the atmospheric light chromaticity of the rendered fog image is defined as:

$$\alpha^c_{r,ch} = \frac{A[\hat{x}^f_{r,ch}]}{A[\hat{x}^f_{r,R}] + A[\hat{x}^f_{r,G}] + A[\hat{x}^f_{r,B}]}, \quad (19)$$

where $A[.]$ is the function that obtains the chromaticity or color of the atmospheric light. This function is basically a color constancy function, hence any color constancy algorithm can be used [10]. In our implementation, to obtain the atmospheric light chromaticity from fog images, we simply use the brightest patch assumption [40].

Therefore, we define our hazeline loss, which is based on the collinearity in the chromaticity space as:

$$\mathcal{L}_{\text{HL}^c_r}(E_c, D_f) = \mathbb{E}_{x^c_r}\big[1 - \frac{(\sigma^c_r - \alpha^c_r)\cdot(\gamma^c_r - \alpha^c_r)}{\|(\sigma^c_r - \alpha^c_r)\|\|(\gamma^c_r - \alpha^c_r)\|}\big]. \quad (20)$$

Like the discriminative loss, while updating the weights of E_c, D_f, we keep other modules frozen.

3.5. Real Fog Data Training Stage

In this stage, we use the real fog images without optical flow ground-truths and without clean-image ground-truths to train the network. As shown in Fig. 2, module E_f takes the fog images, x^f_{r1}, x^f_{r2}, as the input and generate features, which are used by D_{of} to predict the optical flow:

$$\widehat{of}^{\,f} = D_{of}[(E_f[x^f_{r1}], E_f[x^f_{r2}]). \quad (21)$$

D_{of} can handle fog images, since it was trained in the previous stage (Sec. 3.4) using the rendered fog images. At the same time, we transform the input fog images, x^f_{r1}, x^f_{r2} to clean images, $\hat{x}^c_{r1}, \hat{x}^c_{r1}$, respectively:

$$\hat{x}^c_{r1} = D_c[E_f[x^f_{r1}]], \quad (22)$$

$$\hat{x}^c_{r2} = D_c[E_f[x^f_{r2}]]. \quad (23)$$

The transformation modules D_c, E_f had been initially trained in the previous stage as well. From the rendered clean images, $\hat{x}^c_{r1}, \hat{x}^c_{r2}$, we transform them further to obtain the rendered fog images, $\hat{\hat{x}}^f_{r1}, \hat{\hat{x}}^f_{r2}$, respectively:

$$\hat{\hat{x}}^f_{r1} = D_f[E_c[\hat{x}^c_{r1}]], \quad (24)$$

$$\hat{\hat{x}}^f_{r2} = D_f[E_c[\hat{x}^c_{r2}]]. \quad (25)$$

We also compute the optical flow from the rendered clean images, $\hat{x}^c_{r1}, \hat{x}^c_{r2}$:

$$\widehat{\widehat{of^c}} = D_{of}[\hat{x}^c_{r1}, \hat{x}^c_{r2}]. \quad (26)$$

Like in the previous stage, to train the network, we use all the losses we defined in Sec. 3.4, except for the EPE loss. In this training stage, we still compare the EPE loss between $\widehat{of}^f$ and $\widehat{\widehat{of^c}}$, which we call the optical flow consistency loss [36]. However, the goal is no longer for estimating flow accurately, but for the two encoders to extract proper domain adaptive features. Thus, during the backpropagation of this loss, only two encoders E_c and E_f are updated, and the rest are kept frozen.

3.6. Photometric Consistency Map

In the second training stage, we use the rendered clean images, rendered fog images, and estimated optical flow together to train our network. However, the estimated optical flow might still be inaccurate, which can affect the learning process of the whole network. To address this problem, we generate a binary mask based on the photometric consistency of the estimated optical flows. The consistency is computed from the two input clean images and their estimated optical flow. The consistency is then binarized into a mask, and then we apply the mask to the EPE loss. This enables us to filter out the inaccurate estimations of optical flow during the backpropagation.

4. Implementation

Our network in total has two encoders, three decoders and two discriminators. Each of the two encoders contains 6 convolution layers. From an input image, each encoder extracts pyramid features at 6 different levels. As a result, the optical flow decoder has a pyramid structure. Its inputs are the five pairs of pyramidal features from a pair of input images. These features are the five deep-layers features extracted by the encoder. The features of each layer are warped based on the previous level of optical flow, and then we compute the cost volume, which is used to estimate optical flow. As for the two decoder for the domain transformation, the input images, first layer features and second layer features are convoluted into the same shape as the third layer features by a convolution layer. Next, these four features with the same shape are concatenated together, and put into ResNet [13]. This ResNet contains six blocks, and its output has the same shape as input. Finally, the deconvolution layers process on the output from ResNet to generate the domain transformation result. The network architecture of each discriminator is similar to that of the PatchGAN discriminator [15], containing five convolution layers.

For training images, we use randomly cropped images of 256x512 resolution. We set the batch size to 3. We use Adam [17] for the optimizers of all the modules, and its parameters, β_1 and β_2, are set 0.5 and 0.999 respectively. The learning rate is set to 0.0002. All the modules are trained from scratch. We collected real clean and real fog images. All contain urban scenes. We use the VKITTI dataset [9] for rendering synthetic fog images for the fully supervised training, as it has both depth maps and optical flow ground-truths. We specifically select the overcast images (with no sunlight) so that the rendered fog images look more realistic. With the available depth maps, we can generate fog images from VKITTI, with random atmospheric light and attenuation coefficient. The fog in synthetic data is generated by following the physics model [18] for fog, expressed in Eq. (16).

5. Experimental Result

For evaluation, we compare our method with the following methods: original FlowNet2 [14], original PWC-Net [38], which are the two state-of-the-art fully supervised methods; and optical flow network in competitive collaboration (CC) [31] and SelfFlow [26], which are two state-of-the-art unsupervised methods; FlowNet2-fog and PWCNet-fog, where we retrain the original FlowNet2 and PWC-Net using our synthetic fog images and their optical flow ground-truths; FlowNet2-defog, PWCNet-defog and CC-defog which are two-stage solutions where we combine a defogging method with the original FlowNet2, PWCNet and CC. The defogging method is Berman et al.'s [2], which is one of the state-of-the-art defogging methods.

We use 2,224 real clean and 2,346 real fog image pairs for training. For evaluation, following [22], we manually annotate 100 real fog image pairs. The annotated optical flow ground-truths are done for some rigid objects using the manual flow annotation method introduced in [24]. We further use 1,770 and 200 randomly sampled image pairs from the vKITTI dataset for training and validation, respectively. We use other 100 image pairs for testing. We render fog in all the images from the vKITTI dataset using the physics model [18], with random atmospheric light and attenuation coefficient (or fog density).

Quantitative Evaluation Since we target real dense fog images in our method and design, we do the evaluation on real images. EPE and "bad pixel" are commonly used metrics

Input Image | Ground Truth | **Our Result** | PWCNet | PWCNet-defog | PWCNet-fog | FlowNet2

Figure 3: Qualitative comparison of our methods with the state-of-the-art methods and their variants on real fog images.

Table 1: Quantitative results on our real dense foggy dataset.

Method	EPE	Bad Pixel	
		$\delta = 3$	$\delta = 5$
CC [31]	7.56	76.61%	45.79%
CC-defog [31]	11.67	72.22%	42.31%
PWCNet [38]	6.36	54.90%	38.47%
PWCNet-defog [38]	6.16	53.98%	38.50%
PWCNet-fog [38]	6.10	56.39%	39.42%
FlowNet2 [14]	4.74	42.06%	26.75%
FlowNet2-defog [14]	4.72	43.12%	26.60%
FlowNet2-fog [14]	5.19	49.66%	31.89%
SelfFlow [26]	6.53	70.92%	56.01%
Ours	**4.32**	**41.26%**	**25.24%**
Ours (no hazeline)	4.82	43.41%	31.60%

Table 2: Quantitative results on synthetic foggy vKITTI dataset.

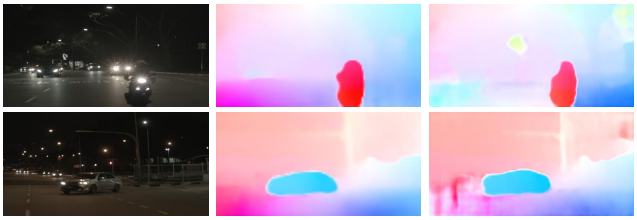
Method	EPE	Bad Pixel	
		$\delta = 1$	$\delta = 3$
CC [31]	7.53	70.54%	51.46%
CC-defog [31]	7.91	65.51%	38.90%
PWCNet	3.23	52.84%	19.01%
PWCNet-defog [38]	3.11	43.93%	18.28%
PWCNet-fog [38]	1.67	34.08%	9.04%
FlowNet2 [14]	5.92	52.42%	30.78%
FlowNet2-defog [14]	5.43	50.05%	28.80%
FlowNet2-fog [14]	9.64	73.02%	48.79%
Ours	**1.60**	**28.31%**	**8.45%**

to measure the quality of optical flow. The definition of "bad pixel" follows to that of the KITTI dataset [27]. Since the flow ground-truths in our evaluation are manually annotated, they might be inaccurate. To account for this, following the KITTI dataset [27], we compute "bad pixel" with its threshold parameter $\delta = \{3, 5\}$, to allow for an inaccuracy of 3-5 pixels. Table 1 shows the evaluation result on our manually annotated real fog images. Our method has the best performance in terms of both EPE value and "bad pixel" numbers. Table 2 shows the results on synthetic fog images from the vKITTI dataset. Since for synthetic fog images, we have accurate dense flow ground-truths, we compute "bad pixel" with $\delta = \{1, 3\}$. While this is not our target, the evaluation shows that our method has comparable performance as the naive solutions.

Qualitative Evaluation Fig. 3 shows the qualitative results on real fog images. The first column shows the first image of the input image pair. All ground-truths in the second column are labeled manually by selecting rigid objects. The third column shows our results, and the other columns show the results of the baseline methods. As can be seen, our method in general performs better than the baseline methods, confirming our quantitative evaluation on the same data. Fig. 4 shows the qualitative defogging results on real foggy images. We compare our method with the state of the art of non-learning method Berman et al. [2] and learning-based method EPDN [29]. Although defogging is not our

Input Image **Our Result** Berman et al. [2] EPDN [29]

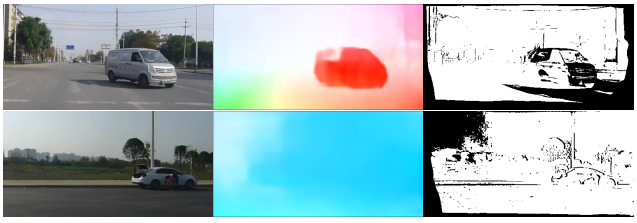
Figure 4: Qualitative defogging results on real foggy images. Although defogging is not our main target, we can observe that our method generates less artifacts than the state of the art methods do.



Input Image **Our Result** PWCNet [38]

Figure 5: Qualitative methods on real nighttime images. As the results show, our method is not limited to fog, but can also work robustly for a different domain, such as nighttime.

Figure 6: In each row, images from left-to-right show the input clean image, and the corresponding rendered fog images with and without the hazeline loss. The hazeline loss constrains our rendered fog images to avoid having fake colors.



Input Clean Image Estiamted Flow Consistency map

Figure 7: The photometric consistency masks correctly indicate the wrong flow estimations.

main target, we can observe that our method generates less artifacts.

While our method is designed for flow estimation under dense fog, we show that our method can also be applied to other domains, such as nighttime. For this, we use our entire training procedure as is, except for the hazeline loss described in Eq. (20). The results are shown in Fig. 5.

6. Ablation Study

Fig. 6 shows the efficacy of our hazeline loss. It can constrain the color shifting between the clean images and the rendered fog images. We can observe that better fog images are generated by using the constraint. Table 1 shows the performance with and without the hazeline loss. Without the hazeline loss, our performance drops by 0.5 for EPE and 2-6% on "bad pixel" rate.

Fig. 7 shows the binary photometric consistency masks. In the first row, our estimated optical flow has error on the minibus back window, and the mask can clearly show that area is inconsistent (black indicates inconsistent predictions, and white indicates consistent predictions). In the second row, the scene is static and the camera is moving. The optical flow is only generated by ego-motion. The estimated optical flow observably has errors on the left top corner. Our consistency mask also indicates the same. The consistency mask and setting proper hyper-parameters (Sec. 4) are important for training stabilization. In our experiments, we find that the training loss can fail to converge if the consistency mask is not used. We also check the efficacy of the domain transformation module. We observe that without this module (i.e. using only E_f and D_{of} in our network in Fig. 2), the performance of our method drops by 1.99 for EPE on real fog images.

7. Conclusion

In this paper, we have proposed a semi-supervised learning method to estimate optical flow from dense fog images. We design a multi-task network that combines domain transformation and optical flow estimation. Our network learns from both synthetic and real data. The synthetic data is used to train our network in a supervised manner, and the real data is used in an unsupervised manner. Our experimental results show the effectiveness of our method, which outperforms the state-of-the-art methods.

References

[1] Dana Berman, Shai Avidan, et al. Non-local image dehazing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1674–1682, 2016. 1, 5

[2] D. Berman, T. Treibitz, and S. Avidan. Single image dehazing using haze-lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. 2, 6, 7, 8

[3] Mario Bijelic, Tobias Gruber, and Werner Ritter. A benchmark for lidar sensors in fog: Is detection breaking down? In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 760–767. IEEE, 2018. 1, 3

[4] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016. 3

[5] Dengxin Dai, Christos Sakaridis, Simon Hecker, and Luc Van Gool. Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding. *International Journal of Computer Vision*, 2019. 2

[6] A. Dosovitskiy, P. Fischer, E. Ilg, , V. Golkov, P. Häusser, C. Hazırbaş, V. Golkov, P. Smagt, D. Cremers, , and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 1, 2

[7] Raanan Fattal. Single image dehazing. *ACM transactions on graphics (TOG)*, 27(3):72, 2008. 1

[8] Denis Fortun, Patrick Bouthemy, and Charles Kervrann. Optical flow modeling and computation: a survey. *Computer Vision and Image Understanding*, 134:1–21, 2015. 2

[9] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 6

[10] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer. Computational color constancy: Survey and experiments. *IEEE Transactions on Image Processing*, 20(9):2475–2489, 2011. 5

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 3, 4, 5

[12] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2011. 1, 5

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[14] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *CoRR*, abs/1612.01925, 2016. 1, 2, 6, 7

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2, 6

[16] J Yu Jason, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016. 2

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 6

[18] Harald Koschmieder. Theorie der horizontalen sichtweite. *Beitrage zur Physik der freien Atmosphare*, pages 33–53, 1924. 1, 6

[19] Wei-Sheng Lai, Jia-Bin Huang, and Ming-Hsuan Yang. Semi-supervised learning for optical flow with generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 354–364, 2017. 2

[20] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4770–4778, 2017. 3

[21] Runde Li, Jinshan Pan, Zechao Li, and Jinhui Tang. Single image dehazing via conditional generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8202–8211, 2018. 1, 3

[22] Ruoteng Li, Robby T Tan, Loong-Fah Cheong, Angelica I Aviles-Rivero, Qingnan Fan, and Carola-Bibiane Schonlieb. Rainflow: Optical flow under rain streaks and rain veiling effect. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7304–7313, 2019. 6

[23] Yu Li, Shaodi You, Michael S Brown, and Robby T Tan. Haze visibility enhancement: A survey and quantitative benchmarking. *Computer Vision and Image Understanding*, 165:1–16, 2017. 3

[24] Ce Liu, William T Freeman, Edward H Adelson, and Yair Weiss. Human-assisted motion annotation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 6

[25] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017. 2, 5

[26] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4571–4580, 2019. 2, 6, 7

[27] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 7

[28] József Molnár, Dmitry Chetverikov, and Sándor Fazekas. Illumination-robust variational optical flow using cross-correlation. *Computer Vision and Image Understanding*, 114(10):1104–1114, 2010. 2

[29] Yanyun Qu, Yizi Chen, Jingying Huang, and Yuan Xie. Enhanced pix2pix dehazing network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7, 8

[30] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. *CoRR*, abs/1611.00850, 2016. 1, 3

[31] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12240–12249, 2019. 2, 6, 7

[32] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Single image dehazing via multi-scale convolutional neural networks. In *European conference on computer vision*, pages 154–169. Springer, 2016. 3

[33] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 2

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[35] Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In *European Conference on Computer Vision (ECCV)*, pages 707–724, 2018. 2

[36] Aashish Sharma, Robby T Tan, and Loong-Fah Cheong. Depth estimation in nighttime using stereo-consistent cyclic translations. *arXiv preprint arXiv:1909.13701*, 2019. 2, 5, 6

[37] Matan Sulami, Itamar Glatzer, Raanan Fattal, and Mike Werman. Automatic recovery of the atmospheric light in hazy images. In *2014 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11. IEEE, 2014. 5

[38] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 1, 2, 3, 6, 7, 8

[39] Robby T Tan. Visibility in bad weather from a single image. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 1, 5

[40] Shoji Tominaga, Satoru Ebisui, and Brian A Wandell. Scene illuminant classification: brighter is better. *JOSA a*, 18(1):55–64, 2001. 5

[41] Yang Wang, Peng Wang, Zhenheng Yang, Chenxu Luo, Yi Yang, and Wei Xu. Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8071–8081, 2019. 2

[42] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. DeepFlow: Large displacement optical flow with deep matching. In *IEEE Intenational Conference on Computer Vision (ICCV)*, Sydney, Australia, Dec. 2013. 1

[43] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate Optical Flow via Direct Cost Volume Processing. In *CVPR*, 2017. 1

[44] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018. 2

[45] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2018. 1

[46] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019. 2

[47] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017. 2, 5