RainFlow: Optical Flow under Rain Streaks and Rain Veiling Effect*

Ruoteng Li[1], Robby T. Tan[1,2], Loong-Fah Cheong[1], Angelica I. Aviles-Rivero[3], Qingnan Fan [4], and Carola-Bibiane Schönlieb[3]

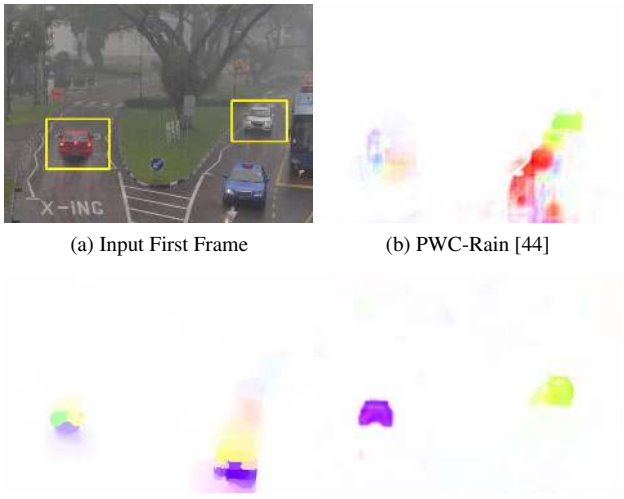
[1]National University of Singapore
[2]Yale-NUS College
[3]University of Cambridge
[4]Stanford University

Abstract

Optical flow in heavy rainy scenes is challenging due to the presence of both rain steaks and rain veiling effect, which break the existing optical flow constraints. Concerning this, we propose a deep-learning based optical flow method designed to handle heavy rain. We introduce a feature multiplier in our network that transforms the features of an image affected by the rain veiling effect into features that are less affected by it, which we call veiling-invariant features. We establish a new mapping operation in the feature space to produce streak-invariant features. The operation is based on a feature pyramid structure of the input images, and the basic idea is to preserve the chromatic features of the background scenes while canceling the rain-streak patterns. Both the veiling-invariant and streak-invariant features are computed and optimized automatically based on the the accuracy of our optical flow estimation. Our network is end-to-end, and handles both rain streaks and the veiling effect in an integrated framework. Extensive experiments show the effectiveness of our method, which outperforms the state of the art method and other baseline methods. We also show that our network can robustly maintain good performance on clean (no rain) images even though it is trained under rain image data. [1]

1. Introduction

Existing optical flow methods (e.g. [49, 1, 44, 22, 19, 20, 39]) show accurate and robust performance in several benchmarking datasets [17, 2, 7]. Most of them, however, still face challenges when applied to rain images [26]. We consider addressing the problem of optical flow in rainy scenes is important, since more and more vision-based systems, which require motion information, are deployed in outdoor environments. Most of them have to work in any weather condition and rain is the most adverse weather phenomenon [40] that occurs frequently in the real world.



(a) Input First Frame (b) PWC-Rain [44]

(c) Robust Flow [26] (d) Ours

Figure 1: An example of our algorithm compared with Robust Flow [26] and PWC-Net [44] on real rain image input. Moving objects are indicated in the yellow boxes.

There are two main properties of rain, particularly heavy rain, which causes existing optical flow methods to be erroneous: rain streaks and the rain veiling effect. Rain streaks occlude the background scene, and appear in different locations in different input frames, and thus induce violation to the brightness constancy constraint (BCC). Rain streaks

*This work is supported by the DIRP Grant R-263-000-C46-232. R.T. Tan's research is supported in part by Yale-NUS College Start-Up Grant.

[1]The code is available at `https://github.com/liruoteng/RainFlow`

also render spurious gradients due to the specular reflection of individual streaks, consequently causing violation to the gradient constancy constraint (GCC). Both the BCC and GCC are the core assumptions of optical flow methods. Hence, existing variational methods [18, 6, 42], patchmatch methods [28, 19], and even some deep learning methods [10, 22, 44, 37, 49] cannot perform adequately in rainy scenes. The rain veiling effect refers to the atmospheric conditions visually similar to fog, which is attributed to light scattering by densely accumulated rain droplets. It occurs particularly in heavy rain. It washes out the background colors and the overall image contrast, making the BCC- and GCC-based methods more susceptible to the aforementioned violations and any further noise [45].

In this paper, our goal is to estimate optical flow from rain images without being affected by the appearance of rain streaks and the rain veiling effect. Particularly, we target heavy rain images, where both rain streaks and the rain veiling effect are present substantially. To accomplish the goal, we propose a deep learning method that requires synthetic rain images and the corresponding optical flow maps to train our network. Our optical flow computation is based on the cost volume (e.g., [49, 44]). Hence, to have a robust optical flow from heavy rain images, we need to ensure that our cost volume is robust to both rain streaks and the rain veiling effect. There are two key ideas in our method.

First, to deal with the loss of contrast issue posed by the rain veiling effect, we compute the cost volume from a feature representation instead of directly from the input rain images. This feature representation is less affected by the rain veiling effect, and we call it veiling-invariant[2] features. The veiling-invariant features are computed by multiplying a feature multiplier with an input image features. The feature multiplier acts as contrast enhancement, which boosts the contrast of the input image features even in the presence of rain veiling effect. It encodes both intensity and depth information from coarse to fine scale. We consider that this encoding allows the contrast to be restored in a depth-aware and scale-dependent manner, thereby better preserving the integrity of the various constancy constraints.

Second, unlike existing methods (e.g. [34, 26]) that handcraft an invariant representation to deal with rain streaks or other artifacts, we propose a rain-streak-invariant features that are automatically learned by our network. To achieve this, our network generates RGB chromatic features, and then transform them into features that are less affected by rain streaks, which we call streak-invariant features. The basic motivation of the transformation is that in an image, rain streaks appear in RGB channels identically. Thus, if we subtract one channel from the other, rain-streak will be cancelled [26]. However, instead of applying the subtraction operation in the image domain, we apply it in feature domain, of which further details and motivations are provided in the ensuing sections. Both the feature multiplier and the streak-invariant features are computed and optimized based on the accuracy of our optical flow estimation. After obtaining the features that are less affected by the rain veiling effect and rain-streaks, we then compute the cost volume before estimating optical flow.

As a summary, in addressing the problem of optical flow estimation from heavy rain images, we make the following contributions:

- We introduce veiling-invariant features, which are less affected by the rain veiling effect. These features are generated using a feature multiplier in the feature domain. The feature multiplier can enhance the contrast of the features in a depth-aware manner, making our features robust to the rain veiling effect.
- We propose a data-driven scheme to learn streaks-invariant features. The ability to automatically learn nonlinear, spatially varying and streak-invariant features is important for coping with the complex perturbations caused by dense rain streaks.
- We propose an integrated and end-to-end framework of optical flow estimation that can handle simultaneously both rain streaks and the rain veiling effect, which are the attributes of heavy rain.

Our experimental results show that our method outperforms the state of the art method and other baselines both qualitatively and quantitatively.

2. Related Works

Most existing deraining methods, including single image based deraining [55, 9, 25, 27, 12, 38, 46, 52] and video based deraining [56, 16, 3, 5, 31, 24, 9, 25, 8, 41], focus on rain streaks removal. These methods do not consider the appearance of rain veiling effect, and hence can only work for relatively light rain scenarios. Yang et al. [51] develop a multi-task network for rain detection and removal, integrating a fog/haze removal module to handle rain veiling effect. However, since the deraining process is performed frame-by-frame independently, the derained output does not guarantee the photo-consistency between consecutive frames, to the detriment of optical flow computation.

Since Horn and Schunck's classic work [18], a large number of variational optical flow approaches have been proposed, examples of which include [4, 6, 30, 42]. Readers are referred to [11] for a recent survey on this topic. As real-world images or video sequences usually contain a certain level of noise and outliers, to this aim, several robust methods have been proposed [47, 48, 30, 50, 35]. While these methods can deal with a moderate amount of image corruptions such as drizzles and light rain, they tend to fail under

[2]The invariant is used in the sense of strongly (not strictly) invariant.

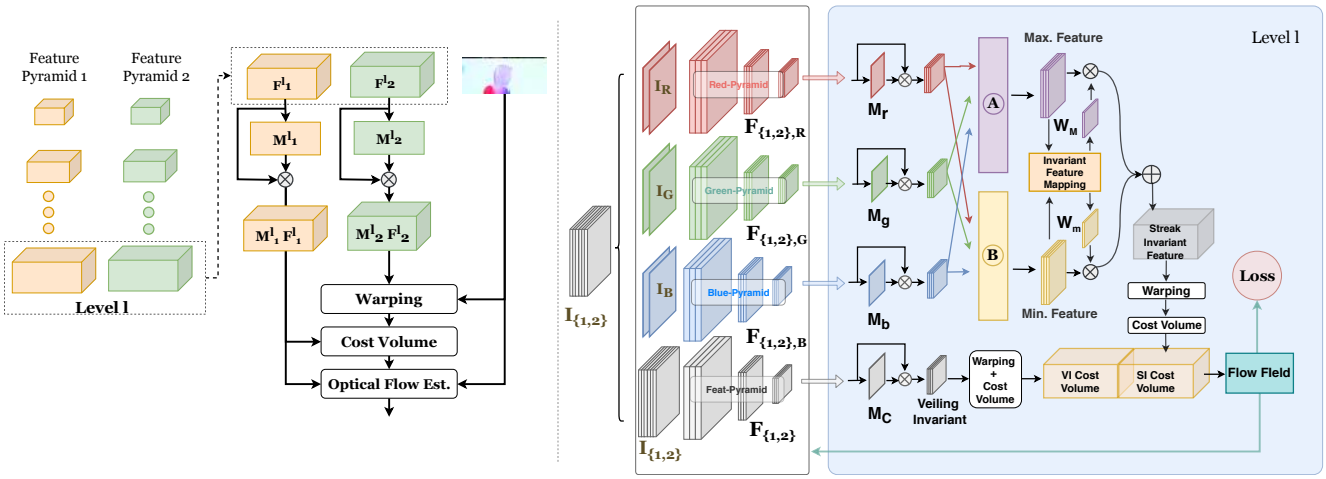


Figure 2: Left: Detailed structure for extracting the feature multiplier $\mathbf{M}$ when computing optical flow at each pyramidal level. Right: the architecture for the full solution. Layer A and B are global maximum and global minimum operation respectively. VI stands for veiling-invariant, and SI stands for streak-invariant.

heavy rain scenarios, which contain strong rain veiling effect and rain streak. Another line of research uses the well-known HSV and the $r\phi\theta$ color space to obtain features that are invariant to illumination changes (see a review in [34]). However, this is not specifically designed to handle rain, and thus does not perform adequately. Li et al. [26] propose a robust optical flow method based on the residue channel that is invariant to rain streaks. However, the spatially-uniform residue image operation resulted in missed motion for objects of comparable size to rain streaks. Moreover, the residue channel is handcrafted, and whether it is an optimal representation for computing optical flow is unknown.

Dosovitskiy et al. [10] propose the first CNN based solution for estimating optical flow. Since then, many CNN based methods have been proposed [37, 49, 21, 54]. Ilg et al. [22] build a large CNN model FlowNet2 by stacking a few basic FlowNets and train it in a stage-by-stage manner. The performance of FlowNet2 can compete with the state of the art variational methods. Sun et al. [44] propose a compact but effective network, PWC-Net, that outperforms FlowNet2 and other state of the art methods. PWC-Net elegantly utilize the cost volume computation, which is widely applied in stereo problems. Though all the aforementioned methods perform well on existing normal optical flow benchmarking datasets, they tend to perform poorly on heavy rain scenarios [26].

3. Proposed Method

We design our network by considering heavy rain, where both rain streaks and the rain veiling effect can have a substantial presence. Our design is thus driven by these two rain components, and the following discussion first focuses on how we deal with each of them. Subsequently, we discuss the integration of our solutions in one framework.

3.1. Optical Flow under Rain Veiling Effect

The left side of Fig. 2 shows our network in dealing with the rain veiling effect. Our backbone network that generates image features is an L-level feature pyramid encoder [44]. $\mathbf{F}^l_{\{1,2\}} = \{\mathbf{F}^l_1, \mathbf{F}^l_2\}$ are the image features at level l of the pyramid that represent the features associated with images 1 and 2, respectively. The bottom level of the pyramid $\mathbf{F}^0_1 = \mathbf{I}_1$, and $\mathbf{F}^0_2 = \mathbf{I}_2$, where $\mathbf{I}_1, \mathbf{I}_2$ are the input images.

To tackle the low contrast problem introduced by the rain veiling effect, we introduce an extra 1×1 *Conv-ReLU* layer that takes the extracted features at each pyramid level as the input, and outputs feature multipliers for every level, $\mathbf{M}^l_{\{1,2\}}$:

$$\begin{aligned} \mathbf{M}^l_1 &= \mathsf{Conv}_1(\mathbf{F}^l_1) \\ \mathbf{M}^l_2 &= \mathsf{Conv}_2(\mathbf{F}^l_2). \end{aligned} \tag{1}$$

Having obtained $\mathbf{M}^l_{\{1,2\}}$, we multiply them with $\mathbf{F}^l_{\{1,2\}}$ in an element-wise manner, resulting in $(\mathbf{M}^l_1(x)\mathbf{F}^l_1(x))$ and $(\mathbf{M}^l_2(x)\mathbf{F}^l_2(x))$, which are the veiling-invariant features from images 1 and 2, respectively. These veiling-invariant features are normalized, and then the matching cost can be computed using the following expression:

$$C^l(x,u) = 1 - (\mathbf{M}^l_1(x)\mathbf{F}^l_1(x))^T(\mathbf{M}^l_2(x+u)\mathbf{F}^l_2(x+u)), \tag{2}$$

where u is the estimated flow at level l of the pyramid. To compute the optical flow, we warp one of the veiling-invariant features, and then compute the cost volume using Eq. (2). Note that, since rain images also contain rain streaks, we do not only use the veiling-invariant features

to compute optical flow, but also streak-invariant features, which will be discussed in Sec. 3.2.

Network Design Ideas Our network is inspired by our analysis on the model of the rain veiling effect. The details are as follows. Due to the light scattering by a volume of suspended water droplets, the rain veiling effect is formed, similar to the visual formation of fog [51]. This formation can be modeled using a widely used fog model [53, 29]:

$$\mathbf{I}(x) = \mathbf{J}(x)\alpha(x) + (\mathbf{1} - \alpha(x))A, \tag{3}$$

where $\mathbf{I}(x)$ is the image intensity at pixel location x. $\mathbf{J}$ is the clean background image. α is the transmission map. A is the atmospheric light, which is assumed to be constant across the entire image, since in a rainy scene, the main source of global illumination is the cloudy skylight, which is diffused light.

According to Eq. (3), there are two main factors of degradation: Light attenuation and airlight. Light attenuation is the first term in the equation, i.e., $\mathbf{J}(x)\alpha(x)$, where $\alpha(x)$ reduces the information of the background scene, $\mathbf{J}(x)$, in the input image, $\mathbf{I}$. Airlight, the second term of Eq. (3): $(\mathbf{1} - \alpha(x))A$, is the light scattering by the water droplets into the direction of the camera [36]. Thus, airlight washes out the image, reducing contrast and weakening the BCC.

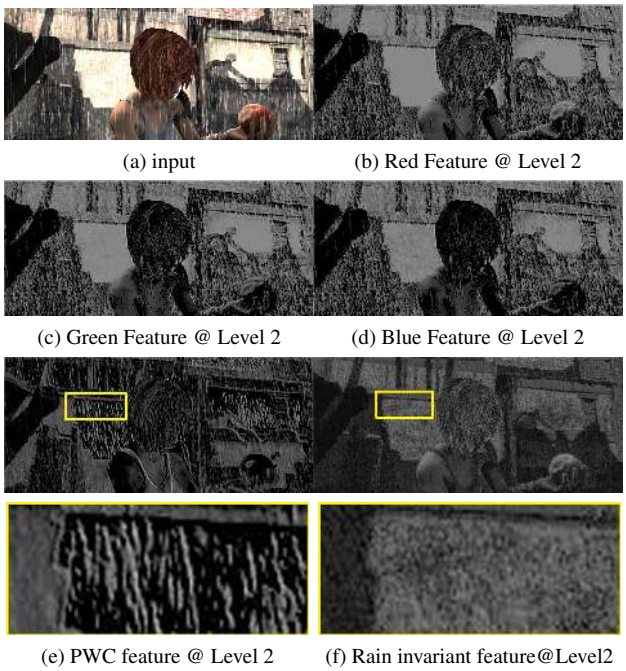
Since $\mathbf{I}_1, \mathbf{I}_2$ are degraded by the rain veiling effect, using them directly or their feature representations to compute optical flow will be sensitive to errors. In contrast, the images of the background scene, $\mathbf{J}_1, \mathbf{J}_2$ are not affected by the rain veiling effect. Thus, we should utilize them or their feature representations. For this, we can reformulate Eq. (3) to express $\mathbf{J}$ in the following form:

$$\mathbf{M}(x)\mathbf{I}(x) = \mathbf{J}(x), \tag{4}$$

where $\mathbf{M}(x) = (\mathbf{I}(x) + \alpha(x)A - A)/\alpha(x)\mathbf{I}(x)$. The presence of this multiplier $\mathbf{M}$, that can generate a veiling-invariant image, inspires us to create a similar multiplier in the feature domain in our network. However, unlike the operation in Eq.(4), our feature multipliers are learned automatically by our network, using the accuracy of our optical flow estimation as our cornerstone.

3.2. Optical Flow under Rain Streaks

Given two input images, $\mathbf{I}_1$ and $\mathbf{I}_2$, we decompose the images into their color channels: R, G, B, denoted as $\mathbf{I}_{1,i}$ and $\mathbf{I}_{2,i}$, where $i \in \{R, G, B\}$. Then, like in the previous section, we create a L-level pyramid of features for each color channel, with the same backbone network. In each level l, we use a *Stride-2* convolution to downsample the features by a factor of 2 to form the next level features. We call the three feature pyramids *chromatic feature pyramids*, which are shown in the right figure of Fig. 2. Each of the chromatic feature pyramids contains features extracted from the color channel of the input rain images, including rain streaks.



(a) input (b) Red Feature @ Level 2 (c) Green Feature @ Level 2 (d) Blue Feature @ Level 2 (e) PWC feature @ Level 2 (f) Rain invariant feature@Level2

Figure 3: The streak-invariant features (f) contains much less rain streaks compared with original PWC feature (g) and R, G, B features (b-d).

At each level l of every chromatic feature pyramid, we apply two operations to extract the maximum and minimum features, and multiply each of these features with some weights, $\mathbf{W}_M$ and $\mathbf{W}_m$:

$$\mathbf{R}_j^l = \mathbf{W}_M \odot \max_i(\mathbf{F}_{j,i}^l) + \mathbf{W}_m \odot \min_i(\mathbf{F}_{j,i}^l), \tag{5}$$

where $i \in \{R, G, B\}$, $j \in 1, 2$ and $\odot$ represents element-wise multiplication. $\mathbf{R}_j^l$ is our rain-streak-invariant features, which are features that are less affected by rain streaks. $\mathbf{W}_M, \mathbf{W}_m$, and $\mathbf{F}_{j,i}^l$ are all learned automatically by our network.

Network Design Ideas The appearance of rain streaks is commonly modeled as a linear combination of the background layer and rain-streak layer (e.g. [53, 12, 32, 23]). Based on this model, Li et al. [26] show that subtracting the minimum color channel from the maximum color channel (i.e., residue channel) will generate a rain-invariant image. Rain streaks are achromatic (white or gray) and appear exactly in the same locations for different RGB color channels, thus subtracting the minimum color channel from the maximum one will cancel the appearance of rain streaks.

While the operation of subtracting the color channel by another color channel in the image space is useful, it can cause damage on the background image since it discards information. Therefore, our idea is to move the operation to the feature domain, so that we can obtain the maximum and minimum feature representations. Moreover, unlike [26], we learn weights via the invariant feature mapping module and apply them to the maximum and minimum representations of the features (Eq. (5)). The values of these weights may be spatially variant, i.e., different for different pixels, and they are learned automatically by the network through the backpropagation process, which uses the optical flow estimation accuracy as the main goal. This spatial variance learns to discard information in a context-dependent manner, e.g., rain-streak free regions should have less information discarded and be less affected by the invariant operation. Through our chromatic feature pyramids and the invariant feature mapping, our network is capable of a more powerful invariant representation than the simple linear operation proposed by [26], and also much more tailored to complex rain streak scenario compared to traditional difference-based invariant such as hue. More importantly, it should still retain sufficient discriminatory information in the computed cost volume, crucial for obtaining robust optical flow under rain streaks. Figure 3 shows some examples of our streak-invariant features.

3.3. Integrated Framework

As heavy rain consists of both the rain veiling effect and rain streaks, to solve them concurrently, we combine the solutions for the rain veiling effect (Sec. 3.1) and rain streaks (Sec. 3.2). The right figure of Fig. 2 shows our integrated network. Given a pair of input images, we create a feature pyramid (at the bottom of the figure) and three chromatic feature pyramids. For each of these feature pyramids, we compute the the feature multipliers ($\mathbf{M}_r$, $\mathbf{M}_g$, $\mathbf{M}_b$, and $\mathbf{M}_C$) and thus obtain the corresponding veiling-invariant features. For the features that focus on the rain-veiling effect (the bottom in the figure), we compute its cost volume, and call it veiling-invariant (VI) cost volume. Meanwhile, for the features focusing on rain streaks, we apply the global maximum (A) and global minimum (B) operations to produce the maximum and minimum features. Subsequently, we run the operation in Eq. (5) to obtain the streak-invariant features, from which we can calculate the streak-invariant (SI) cost volume. We concatenate the veiling-invariant and streak-invariant cost volumes to compute the final optical flow.

Loss Function The loss function of our network is expressed as:

$$\mathcal{L}(\Theta) = \sum_{l}^{L} \alpha_l \sum_{x} (|u^l_\Theta(\mathbf{x}) - u^l_{gt}(\mathbf{x})|_2) + \gamma|\Theta|_2, \quad (6)$$

where Θ represents all the learnable parameters in our network. u_Θ is the predicted optical flow by our network, and u_{gt} is the optical flow ground-truth. $|\cdot|_2$ indicates L2 norm of a vector and the weighting factor $\gamma = 0.0004$ in our experiment. α_l is the learning weight parameter for each pyramid level. We set $\alpha_6 = 0.32$, $\alpha_5 = 0.08$, $\alpha_4 = 0.02$,$\alpha_3 = 0.01$, $\alpha_2 = 0.005$ in practice. The second term regularizes all trainable parameters of the network. As described in the loss function, our network requires only training input data degraded by rain and the ground-truth optical flow. We do not need the corresponding clean (non-rain) images in our method.

4. Implementation

Training Details Unlike many other CNN-based optical flow methods (e.g. [10, 22, 44]), we randomly initialize and train our network using a mixed combination of the FlyingChairs dataset [10] and the downsampled FlyingThings3D dataset [33], instead of separating the two datasets for different training phases. We call this *ChairThingsMix* dataset. Since the average displacement of FlyingThings3D is around 38 pixels, which is higher than the 19 pixels of FlyingChairs, we downsample the FlyingThings3D data to half of its resolution, 270×480 pixels. The image pairs in FlyingThings3D with extreme motion (magnitude larger than 1000 pixels) are excluded. In total, there are 41,732 image pairs in the training set. Since the real rain test dataset has small average motion, we use FlyingChairSDH [22] to construct the ChairThingsMix dataset.

We use the S_{long} learning rate schedule described in [22], starting from 0.0001 and reducing the learning rate by half at 400K, 600K, 800K, 1M iterations with batch size equal to 8. For the data augmentation, we use a simple strategy, including only random translation and random image size scaling. Specifically, the scaling parameter is uniformly sampled from $[0.95, 1.05]$, and the translation parameter (tx, ty) from [-5%, 5%] of the image width w and height h. After data augmentation, we crop 256 ×448 patches as our network input.

During the training, we scale down the flow ground-truths by 20 as suggested in [10, 44]. The downsampled flow ground-truths are sent to each pyramid level. The feature pyramid and chromatic feature pyramids have 6 levels starting from $l_0 = 2$, i.e. the last layer of our network outputs a quarter size of the original flow field. We use bilinear interpolation to upsample the output flow field. Regarding

Table 1: Average EPE results on the FVR-660 and NUS-100 dataset. For derain data, we apply Yang et al's [51] to perform deraining preprocessing

Method	FVR-660		NUS-100		Time (s)
Condition	Rain	Derain	Rain	Derain	Rain
Classic+NL	2.17	2.19	0.49	0.53	47.51
LDOF	2.93	2.98	0.68	0.60	76.00
EpicFlow	4.52	5.50	0.35	0.36	15.00
Robust-Flow	1.76	1.80	0.22	**0.19**	69.94
SpyNet	2.43	2.42	1.41	1.50	0.16
DCFlow	46.71	30.69	0.30	0.30	8.60
FlowNet2	5.73	6.07	0.28	0.30	0.12
FlowNet2-Rain	2.21	2.18	0.42	0.43	0.12
PWC-Net	2.66	2.57	0.49	0.53	**0.02**
PWC-Net-Rain	6.29	6.29	0.87	0.90	**0.02**
RainFlow-Rain	**1.57**	**1.60**	**0.18**	**0.19**	0.03

the cost volume computation, we set the search range to 4 pixels and the kernel size is 1 pixel.

Rain Rendering Details Due to the absence of large-scale real rain sequences with flow ground-truths, we render synthetic rain. Our synthetic rain, containing both rain streaks and the rain veiling effect, is rendered by the following model introduced in [51]:

$$\mathbf{I}(x) = \alpha(x)(\mathbf{J}(x) + \sum_i \mathbf{S}_i(x)) + (\mathbf{1} - \alpha(x))\mathbf{A}, \quad (7)$$

where $\mathbf{I}(x)$ is the image intensity at pixel location $\mathbf{x}$, $\mathbf{J}$ the clean background image, α the transmission map, and $\mathbf{S}_i$ the rain streak layer at the depth-layer i. $\mathbf{A}$ is the atmospheric light, which is assumed to be constant across the entire image, since in a rainy scene, the main source of global illumination is the cloudy skylight, which is diffused light.

For the rendering of rain streaks $\mathbf{S}_i$, we generate photo-realistic rain streaks following Garg et al.'s rain model [15] during the training process. For the FlyingChair dataset, as it has no depth information available, we render the rain veiling effect uniformly across each image. The transmission α is uniformly sampled from the range $[0.3, 1]$. The atmospheric light is uniformly sampled from $[0.4, 1]$. Since FlyingThings3D provides the depth information, we sample the attenuation factor β uniformly from $[3, 5]$ according to $\alpha(x) = \exp^{-\beta D(x)}$, where $D(x)$ is the depth of the scene at location x.

5. Experimental Result

Three datasets are used in our evaluation: 1) Synthetic rain rendered on MPI Sintel [7], KITTI [17] and VKITTI [13] datasets, 2) Hybrid rain of the FVR-660 dataset [26], 3) Real-World rain with human annotated ground-truths, i.e., the NUS-100 dataset [26]. As for the baseline methods, we choose a few conventional methods, i.e. Classic +NL [43], LDOF [6], Epic-Flow, Robust-Flow [26] as well as recent supervised learning methods such as FlowNet2 [22], DCFlow [49], and PWC-Net [44]. For comprehensive and fair comparisons, we train the baseline methods on the same dataset as described in Sec. 4. We indicate those networks trained on rainy data with the suffix "-Rain" (e.g. FlowNet2-Rain, PWC-Rain, etc.). We train these baselines ([22, 44]) according to the training details described in their paper.

Table 2: Average EPE results on the MPI Sintel dataset. We evaluate both rain and clean weather conditions of the two datasets. For the variational methods under rain data, we apply Yang et al.'s [51] to perform deraining preprocessing.

Method	Sintel (train)		VKITTI		KITTI2012
Condition	Clean	Rain	Clean	Rain	Rain
Classic+NL	4.94	7.97	8.06	12.44	9.17
LDOF	4.29	10.68	12.69	19.38	10.17
EpicFlow	2.46	14.92	**4.82**	10.46	6.94
Robust-Flow	4.71	5.46	7.45	11.72	6.65
SpyNet	4.19	9.84	10.21	13.53	11.70
FlowNet2	**2.02**	7.68	6.13	9.12	7.23
FlowNet2-Rain	4.65	6.90	9.586	11.27	8.01
PWC-Net	2.55	14.20	6.73	11.39	7.55
PWC–Net-Rain	4.46	7.26	9.40	9.69	6.41
RainFlow-Rain	2.61	**4.59**	6.90	**8.27**	**5.62**

We test all the baseline methods on the rain-rendered MPI Sintel [7] and KITTI2012 [17] datasets adopted from [26]. We also test all the methods on the VKITTI dataset [14] as it provides all kinds of weather conditions including rain-rendered sequences. All the CNN-based baseline methods are trained on the *ChairThings* (clean and rain) datasets for a fair comparison.

Qualitative Results The qualitative results for the synthetic rain datasets (Sintel, KITTI2012 and VKITTI) are shown in Fig. 4. Real rain results are demonstrated in Fig. 6 and Fig. 7 respectively.

Quantitative Results The quantitative results of the synthetic rain datasets are shown in Table 2. The real rain results are demonstrated in Table 1. From the results shown in the table, our network consistently outperforms all the baseline methods on the synthetic rain datasets. For the clean (no rain) sequences, one can see that most of the current CNN-based optical flow networks face performance degradation on the clean testing datasets when they are trained under the rain data due to the over-fitting problem. However, thanks to the rain-invariant features in our network, our method still produces robust results on both rain and clean testing datasets.

6. Ablation Study

Effectiveness of Multiplier $\mathbf{M}$ To verify the effectiveness of the learnt parameter $\mathbf{M}$, we perform a comparison on PWC-Net and PWC-M, a PWC-based model added with

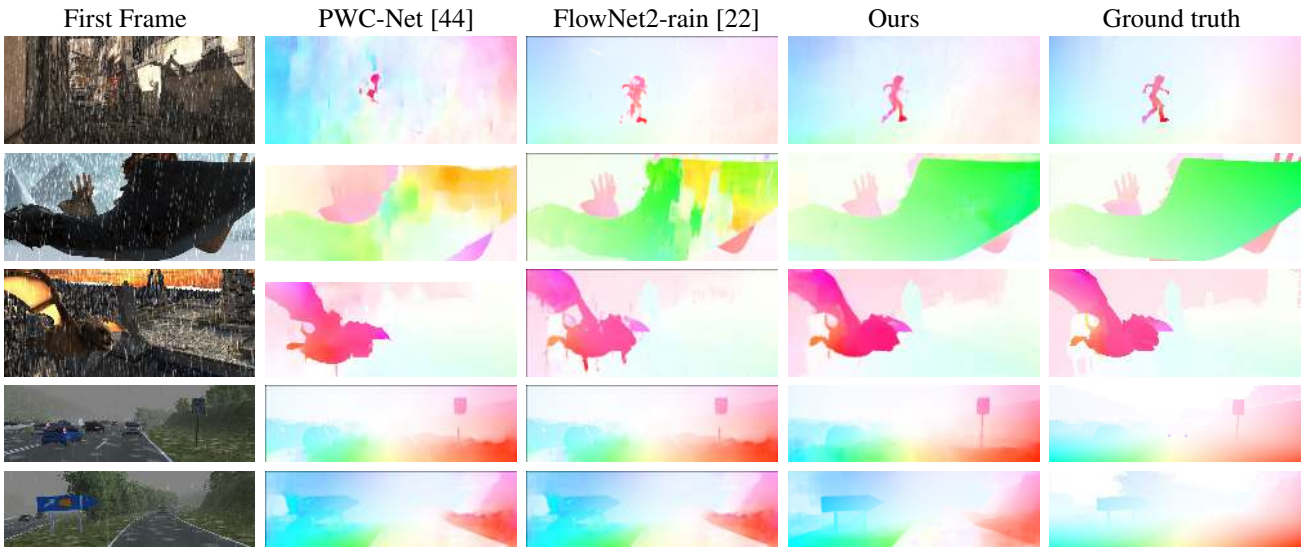
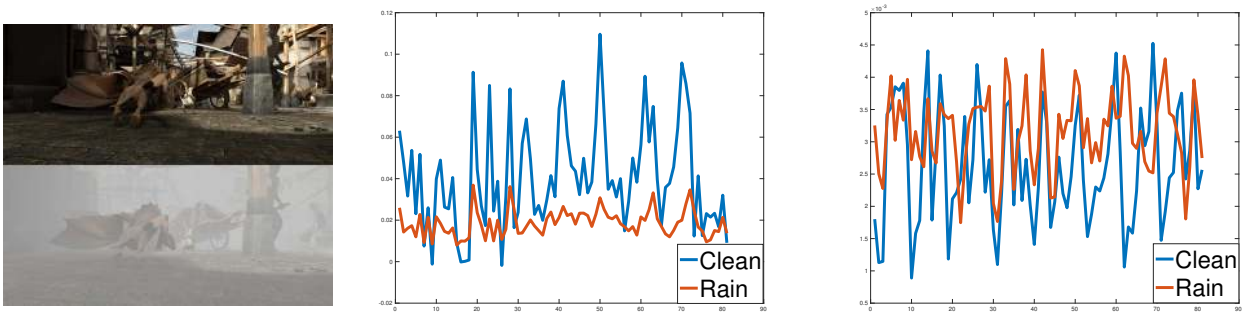


Figure 4: A qualitative comparison of baseline methods and our method on MPI Sintel [7] and VKITTI [13]datasets.



(a) First frame (b) PWC cost volume (c) PWC-M cost volume

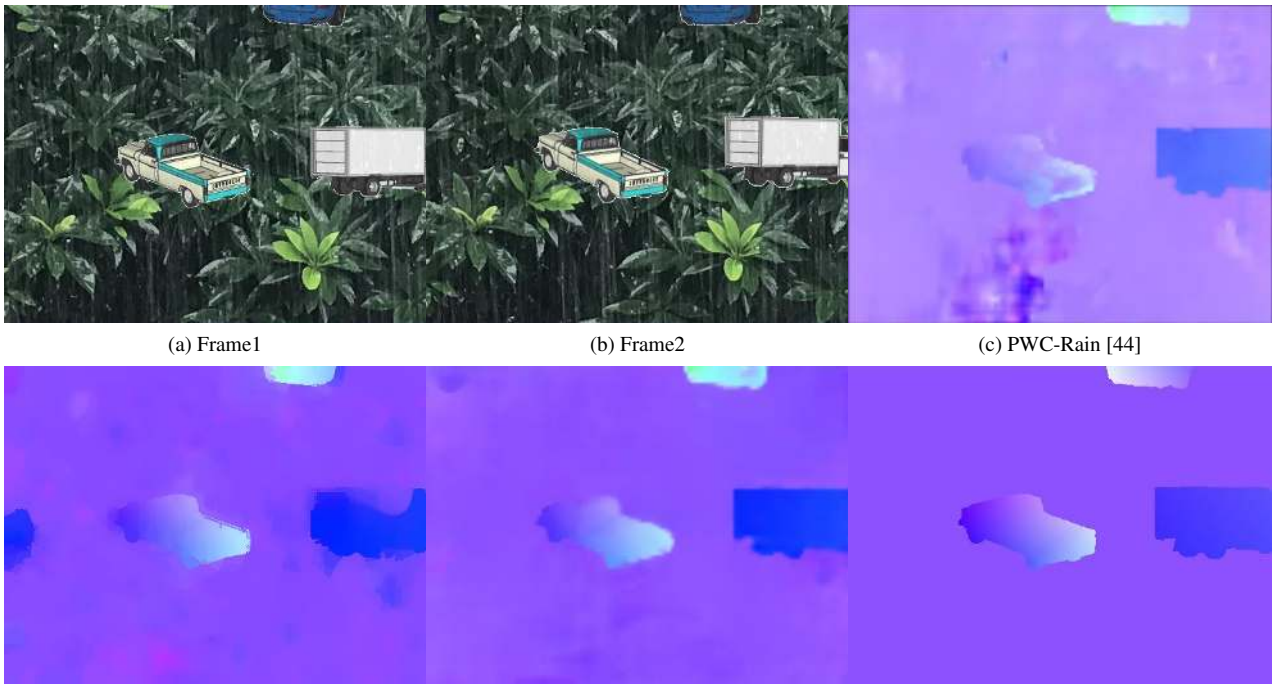
Figure 5: Cost volume analysis of PWC and PWC-M on clean and rain images. (b,c) show the cost volume values of the pixels indicated by the red dots (a). For the graph, x-axis shows the channel index of the cost volume tensor, and y-axis represents the cost volume value.

Table 3: Effectiveness of the feature multiplier $\mathbf{M}$ and the Chromatic Pyramid (ChromPyrd).

Method	Sintel		VKITTI	
Condition	clean	rain	clean	rain
PWC-rain	4.46	7.26	9.40	9.69
Ours w/o **M**	3.67	6.27	8.27	8.96
Ours w/o ChromPyrd	4.29	6.03	6.75	9.38

feature multiplier $\mathbf{M}$ at each level, on the estimated flow on Sintel rendered with a strong rain veiling effect as shown in Table 3. In this experiment, we use PWC-Net as our baseline since it does not have parameter $\mathbf{M}$. We create a model called *PWC-M* by adding multiplier $\mathbf{M}$ learning at each pyramid level of PWC-Net. We train both PWC-M network and PWC network on the same training data described in Sec. 4 and test them on the Sintel dataset rendered with the rain veiling effect only. From the table, we find that the performance of PWC-M outperforms PWC on both rain and clean data. In addition, we also investigate the cost volume on these two network models. In Fig. 5, we plot the cost of a pixel. One can see that the variation of the cost volume of rain input is much smaller than that of clean input for PWC network, whereas with the feature-multiplier $\mathbf{M}$ added to PWC network, the cost volumes of rain input and clean input have similar range of variation. Therefore, the optical flow decoder is able to compute the flow field robustly for both rain sequences and clean sequences.

Effectiveness of Chromatic Pyramids To verify the effectiveness of the chromatic pyramids and invariant feature mapping, we compare PWC network and PWC network with chromatic pyramids and invariant feature map-



(a) Frame1 (b) Frame2 (c) PWC-Rain [44]

(d) Robust Flow [26] (e) Ours (f) Ground Truth

Figure 6: A qualitative comparison of baseline methods and our method on FVR-660 dataset [26].

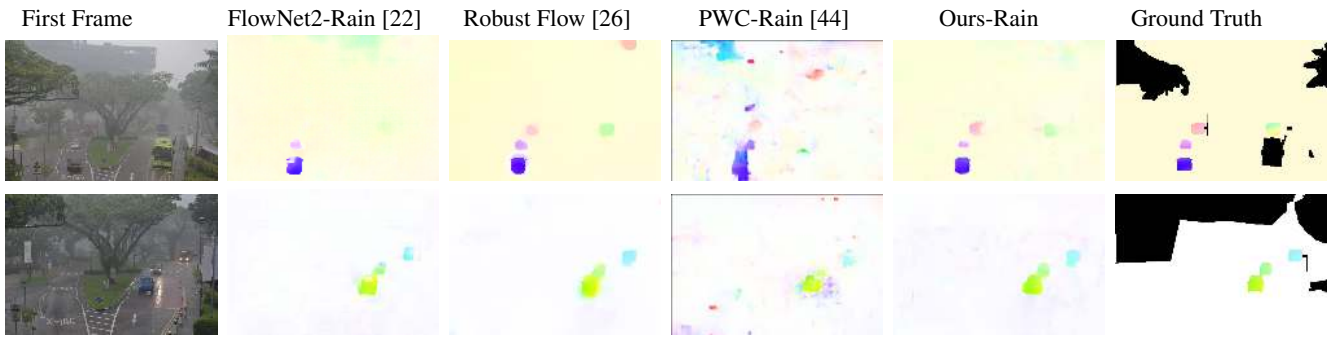


Figure 7: A qualitative comparison of baseline methods and our method on NUS-100 dataset [26].

ping, denoted as PWC-Chromatic. We use Sintel rendered with strong rain streaks and VKITTI datasets for evaluation. The quantitative results are shown in Table 3. PWC-Chromatic is able to outperform PWC network on all the rain datasets with only a marginal increase in the number of parameters needed (i.e. for the invariant feature mapping module). In addition, it also performs better than PWC network on clean datasets. This is because the chromatic feature pyramids and the invariant feature mapping are able to extract more texture-rich features from the background.

7. Conclusion

We present a robust optical flow method that achieves state of the art performance in rainy scenes. To deal with the rain veiling effect, our network learns a contrast-enhancing feature-multiplier $\mathbf{M}$ at each pyramid level so that the cost volume of rainy images is as discriminative as that of a clean image pair. To address the spurious gradients of densely distributed rain streaks, we propose a chromatic feature pyramids that produce a streak-invariant features that are less affected by rain streaks. In addition, our network performance is not at the expense of optical flow estimation on clean sequences even if it is trained under rain conditions. Our experiments demonstrate that our network outperforms all the baselines on all the existing benchmarking datasets.

References

[1] C. Bailer, K. Varanasi, and D. Stricker. Cnn-based patch matching for optical flow with thresholded hinge embedding loss. pages 2710–2719, 07 2017.

[2] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, Mar. 2011.

[3] P. Barnum, T. Kanade, and S. Narasimhan. Spatio-temporal frequency analysis for removing rain and snow from videos. In *Proceedings of the First International Workshop on Photometric Analysis For Computer Vision-PACV 2007*, pages 8–p. INRIA, 2007.

[4] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75 – 104, 1996.

[5] J. Bossu, N. Hautière, and J.-P. Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *International Journal of Computer Vision*, 93(3):348–367, Jul 2011.

[6] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011.

[7] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. *A Naturalistic Open Source Movie for Optical Flow Evaluation*, pages 611–625. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[8] J. Chen and L. Chau. A rain pixel recovery algorithm for videos with highly dynamic scenes. *IEEE Transactions on Image Processing*, 23(3):1097–1104, March 2014.

[9] J. Chen, C.-H. Tan, J. Hou, L.-P. Chau, and H. Li. Robust video content alignment and compensation for rain removal in a cnn framework. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[10] 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.

[11] D. Fortun, P. Bouthemy, and C. Kervrann. Optical flow modeling and computation. *Comput. Vis. Image Underst.*, 134(C):1–21, May 2015.

[12] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

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

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

[15] K. Garg and S. K. Nayar. Photorealistic rendering of rain streaks. *ACM Trans. Graph.*, 25(3):996–1002, July 2006.

[16] K. Garg and S. K. Nayar. Vision and rain. *Int. J. Comput. Vision*, 75(1):3–27, Oct. 2007.

[17] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[18] B. K. P. Horn and B. G. Schunck. Determining optical flow. *ARTIFICAL INTELLIGENCE*, 17:185–203, 1981.

[19] Y. Hu, R. Song, and Y. Li. Efficient coarse-to-fine patch match for large displacement optical flow. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5704–5712, June 2016.

[20] T.-W. Hui, X. Tang, and C. C. Loy. A Lightweight Optical Flow CNN - Revisiting Data Fidelity and Regularization. 2019.

[21] E. Ilg, O. Cicek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[22] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *CoRR*, abs/1612.01925, 2016.

[23] L. W. Kang, C. W. Lin, and Y. H. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 21(4):1742–1755, April 2012.

[24] J. H. Kim, J. Y. Sim, and C. S. Kim. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE Transactions on Image Processing*, 24(9):2658–2670, Sept 2015.

[25] M. Li, Q. Xie, Q. Zhao, W. Wei, S. Gu, J. Tao, and D. Meng. Video rain streak removal by multiscale convolutional sparse coding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[26] R. Li, R. T. Tan, and L.-F. Cheong. Robust optical flow in rainy scenes. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[27] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[28] Y. Li, D. Min, M. S. Brown, M. N. Do, and J. Lu. Spm-bp: Sped-up patchmatch belief propagation for continuous mrfs. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4006–4014, Dec 2015.

[29] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. Brown. Single image rain streak separation using layer priors. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 2017.

[30] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):978–994, May 2011.

[31] J. Liu, W. Yang, S. Yang, and Z. Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[32] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3397–3405, Dec 2015.

[33] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134.

[34] Y. Mileva, A. Bruhn, and J. Weickert. *Illumination-Robust Variational Optical Flow with Photometric Invariants*, pages 152–162. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[35] M. A. Mohamed, H. A. Rashwan, B. Mertsching, M. A. García, and D. Puig. Illumination-robust optical flow using a local directional pattern. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(9):1499–1508, Sept 2014.

[36] S. G. Narasimhan and S. K. Nayar. Vision and the atmosphere. *International journal of computer vision*, 48(3):233–254, 2002.

[37] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network. *CoRR*, abs/1611.00850, 2016.

[38] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang. Video desnowing and deraining based on matrix decomposition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[39] Z. Ren, O. Gallo, D. Sun, M. Yang, E. B. Sudderth, and J. Kautz. A fusion approach for multi-frame optical flow estimation. *CoRR*, abs/1810.10066, 2018.

[40] S. R. Richter, Z. Hayder, and V. Koltun. Playing for benchmarks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2232–2241, 2017.

[41] V. Santhaseelan and V. K. Asari. Utilizing local phase information to remove rain from video. *International Journal of Computer Vision*, 112(1):71–89, Mar 2015.

[42] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2439. IEEE, June 2010.

[43] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2439. IEEE, June 2010.

[44] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

[45] A. Verri and T. Poggio. Motion field and optical flow: Qualitative properties. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(5):490–498, May 1989.

[46] W. Wei, L. Yi, Q. Xie, Q. Zhao, D. Meng, and Z. Xu. Should we encode rain streaks in video as deterministic or stochastic? In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[47] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *IEEE Intenational Conference on Computer Vision (ICCV)*, Sydney, Australia, Dec. 2013.

[48] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006*, pages 211–224, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[49] J. Xu, R. Ranftl, and V. Koltun. Accurate Optical Flow via Direct Cost Volume Processing. In *CVPR*, 2017.

[50] H. Yang, W. Y. Lin, and J. Lu. Daisy filter flow: A generalized discrete approach to dense correspondences. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3406–3413, June 2014.

[51] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Joint rain detection and removal via iterative region dependent multi-task learning. *CoRR*, abs/1609.07769, 2016.

[52] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1694, July 2017.

[53] W. Yang, R. T. Tan, J. Feng, J. Liu, S. Yan, and Z. Guo. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[54] Y. Yang and S. Soatto. Conditional prior networks for optical flow. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[55] H. Zhang and V. M. Patel. Density-aware single image deraining using a multi-stream dense network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[56] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng. Rain removal in video by combining temporal and chromatic properties. In *2006 IEEE International Conference on Multimedia and Expo*, pages 461–464, July 2006.