

Robust Optical Flow in Rainy Scenes*

Ruoteng Li¹, Robby T. Tan^{1,2}, and Loong-Fah Cheong¹

¹ National University of Singapore

² Yale-NUS College

Abstract. Optical flow estimation in rainy scenes is challenging due to degradation caused by rain streaks and rain accumulation, where the latter refers to the poor visibility of remote scenes due to intense rainfall. To resolve the problem, we introduce a residue channel, a single channel (gray) image that is free from rain, and its colored version, a colored-residue image. We propose to utilize these two rain-free images in computing optical flow. To deal with the loss of contrast and the attendant sensitivity to noise, we decompose each of the input images into a piecewise-smooth structure layer and a high-frequency fine-detail texture layer. We combine the colored-residue images and structure layers in a unified objective function, so that the estimation of optical flow can be more robust. Results on both synthetic and real images show that our algorithm outperforms existing methods on different types of rain sequences. To our knowledge, this is the first optical flow method specifically dealing with rain. We also provide an optical flow dataset consisting of both synthetic and real rain images.

Keywords: Optical Flow · Rain · Decomposition · Residue Channel

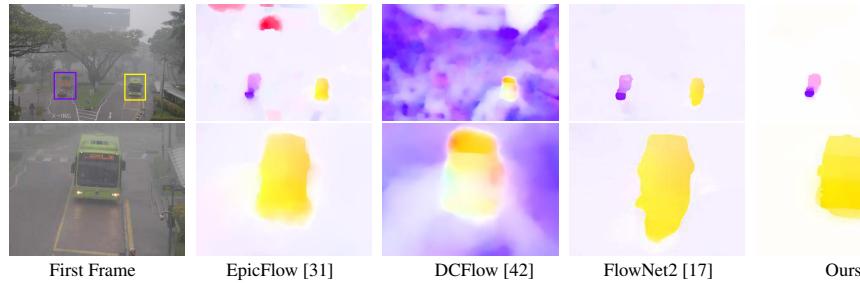


Fig. 1: Optical flow estimation from heavy rain images with a static background and a few moving vehicles. **Top:** Purple and yellow colors indicate upward and downward motions, respectively. **Bottom:** Focusing on the estimated flow of the downward moving vehicle. Our estimation is more robust to rain than those of existing methods. Note, the rain streaks and rain accumulation can be observed by zooming in the input images.

* 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 Introduction

Optical flow methods have been developed for many decades [16,22][36,33,7,38] and achieved significant results in terms of accuracy and robustness[40,1,43,32,17,29]. They are shown to generally work when applied to outdoor scenes in clear daylight, but tend to be erroneous in bad weather. In particular, of all the bad weather conditions, rain has the most marked detrimental impact on performance [28]. To our knowledge, no methods have been proposed to handle optical flow estimation in rainy scenes. Addressing this problem is important, since more and more vision systems are deployed in outdoor scenes, and rain is an inevitable natural phenomenon or even an everyday occurrence in some regions of the world. In this paper, we develop an optical flow algorithm that can handle heavy rain with apparent rain streaks and a fog-like rain accumulation effect.

The challenge of estimating optical flow in rainy scenes can be categorized into two problems. One problem refers to rain streaks, which due to their dynamic nature, appear in different locations from frame to frame, thus causing violation to the brightness constancy constraint (BCC) and the gradient constancy constraint (GCC). The other problem refers to rain-streak accumulation, where rain streaks are accumulated along the line of sight in such a way that we can no longer see the individual streaks (visually similar to fog). Images affected by the rain accumulation generally suffer from low contrast and weakened background information. Under torrential downpour, the second problem is severe enough to warrant a special mechanism to come to grips with the issue.

A direct solution is to apply a deraining method, either a video based (e.g., [14,3]) or single-image based deraining method (e.g. [19,23,47]), before optical flow computation. However, most of the video-based deraining methods are designed only for rain streaks removal and assume static background, whereas the existing single-image based deraining methods process each frame independently, and therefore consistency across frames cannot be guaranteed. Moreover, most of the deraining methods introduce artifacts, such as blur around rain streak regions, high frequency texture loss, image color change, etc. These artifacts are also inconsistent in their appearance throughout an image sequence, thus rendering the BCC and GCC invalid.

To achieve our goal, there are several key ideas in our method. First, we introduce a residue channel, a gray image that is free from both rain streaks and rain accumulation. This rain free can be generated after ensuring that the rain-streak and rain-accumulation terms in our model are achromatic (colorless). The residue channel, however, can cause some color boundaries to disappear, making the optical flow computation deprived of this important information. To resolve this, we then introduce a colored-residue image, which is the colored version of the residue channel and thus is also free from rain streak and accumulation. Yet, there is another problem with both the residue channel and the colored-residue image, namely, when the background is achromatic or the rain accumulation is considerably thick, then the affected regions become dark due to the subtraction operation in the residue image formation, depriving the optical flow estimation of any intensity variation information in the achromatic channel. Generally, images of rainy scenes already suffer from low contrast; this is further compounded by the residue operation such that the signal-to-noise ratio is further suppressed, reaching the nadir in the aforementioned dark regions.

To handle the resultant noise sensitivity, our solution is to perform a structure-texture decomposition on each of the input images and use the extracted structure layers to provide a further regularizing influence on the optical flow estimation. The underlying rationales are twofold: (1) The structure-texture decomposition acts as a denoiser, moving noise, rain streaks, and fine textures in the scene to the texture layer. While the structure layer necessarily loses out some fine texture information, it provides a stabilizing influence on the detailed flow information coming from the colored-residue image. (2) For the regions in the colored-residue image that are dark, the information coming from the structure layer is all that we have; even though it is admittedly lacking in details and might be somewhat inaccurate (since in dealing with rain, the structure layer extraction does not make use of the underlying physics of the rain formation process). Finally, by combining the colored-residue images and structure layers in one objective function, we make sure that the structure-texture variational denoising is done in a way consistent across images (critical for optical flow estimation), and the decomposition can also benefit from the redundancy coming from the multiple frames. As a result, we can compute the flow from rain images robustly.

Particularly with respect to optical flow computation in rainy scenes, our contributions are: (1) Introducing the residue channel and colored-residue image that are both free from rain streaks and rain accumulation, (2) proposing an objective function and its optimization mechanism that combine the colored-residue images and piecewise-smooth structure layers, (3) providing a real rain optical flow benchmark containing both synthesized motion (660 sequences) and real motion (100 sequences) to the public. Note that, in this paper we do not address raindrops attached to the camera lens. We assume that the camera is well protected from raindrops (e.g. placing the camera under a shelter, or using a special hardware like Spintec, which can deflect rain from the camera).

2 Related Work

Optical flow algorithms that are robust to noise and outliers have been studied for a long time (e.g., [5,4,39,34]). While these techniques may be able to handle a moderate amount of corruptions such as those brought about by a drizzle [41,21,45,26], they are unlikely to prevail against the heavy corruptions caused by a torrential downpour. Compounding these issues is the loss of contrast caused by rain accumulation, it causes both the BCC and GCC to be highly susceptible to noise.

One of the popular practices in optical flow estimation is to perform structure-texture decomposition. [37][35] and then use the texture layer for optical flow computation. However, for rainy scenes, rain streaks and significant noise will appear in the texture layer and compromise the utility of the texture layer for flow estimation. Our method relies more on the structure rather than the texture layer. Yang et al. [46] propose a double-layer decomposition framework for estimating optical flow of reflective surfaces. The method decomposes transmission (foreground) layer and reflection (background) layer, and then computes the optical flow of each layer, assuming both layers follow the sparse gradient distribution of natural images. However, this algorithm cannot be applied to rain images, since the assumption does not hold for rain streaks and

accumulation. Our proposed colored-residue image belongs to the class of color space transformation methods such as [25] to render invariance against perturbation. However, the well-known HSV and $r\phi\theta$ color space approaches do not result in measures that are invariant under rain, and hence cannot be directly applied to rain images.

It is beyond the scope of this paper to offer a comprehensive review of the immense optical flow literature, but the emerging deep learning approach certainly deserves a mention. A few deep learning methods (e.g., [11,10,17,30]) are proposed to estimate flow, but these methods are meant for optical flow estimation under clear scenes. Moreover, these methods are heavily optimized over a lot of training data with ground truths. Unfortunately, obtaining the optical flow ground-truths for rainy scenes is considerably intractable. In contrast, our method leverages on the physics of the image formation process. Theoretically, our rain streak formation model and the residue channel idea are applicable to snow and sleet too; our approach thus offers a much more parsimonious solution to a range of problems posed by different weather phenomena.

3 Residue Channel

3.1 Rain Streak Image Formation

The appearance of rain streaks is the result of raindrop movement during the camera exposure[12]. If we assume the exposure time is T and the elapsed time while a raindrop is passing through a pixel \mathbf{x} is τ , the intensity captured by the CCD sensor can be described as a linear combination of the raindrop's time-average radiance \bar{E}_{rs} and the background radiance E_{bg} :

$$E(\mathbf{x}) = \tau \bar{E}_{rs}(\mathbf{x}) + (T - \tau) E_{bg}(\mathbf{x}), \quad (1)$$

where $\bar{E}_{rs} = \frac{1}{\tau} \int_0^\tau E_{rs} dt$, $0 \leq \tau \leq T$. E_{rs} is the radiance of the raindrop at a particular time. The value of E_{rs} is determined by the raindrop's specular and internal reflections, in addition to the refracted light, where their proportions depends on the direction of the light rays relative to the raindrop, the camera viewing direction, and the shape of the raindrop [15].

Most cameras have spectral sensitivities (a.k.a. camera RGB filters) to produce coloured images. Considering this, we express the colored-image intensity of a rain-streak image as:

$$\tilde{\mathbf{I}}(\mathbf{x}) = \tau \int_{\Omega} \bar{E}_{rs}(\mathbf{x}, \lambda) q_c(\lambda) d\lambda + (T - \tau) \int_{\Omega} E_{bg}(\mathbf{x}, \lambda) q_c(\lambda) d\lambda, \quad (2)$$

where $\tilde{\mathbf{I}} = (\tilde{I}_r, \tilde{I}_g, \tilde{I}_b)^T$ is the color vector representing the colored intensity, λ is the light wavelength, q_c is the camera-spectral-sensitivity distribution function, with index c indicates the RGB color channel. Ω is the range of wavelengths that can be captured by the camera sensitivities. As shown in the appendix, we can express the model as:

$$\tilde{\mathbf{I}}(\mathbf{x}) = \tau \rho_{rs}(\mathbf{x}) L\boldsymbol{\sigma} + (T - \tau) B\boldsymbol{\pi}, \quad (3)$$

where $\mathbf{L} = (L_r, L_g, L_b)^T$ is the color vector of the light brightness, and $\mathbf{B} = (B_r, B_g, B_b)^T$ is the color vector of the background reflection. $L = L_r + L_b + L_g$ and $B = B_r + B_b + B_g$. We define $\sigma = \mathbf{L}/L$ and $\pi = \mathbf{B}/B$, the chromacities of \mathbf{L} and \mathbf{B} , respectively. ρ_{rs} is composed of refraction, specular reflection, and internal reflection coefficients of a raindrop [13]. We assume that ρ_{rs} is independent from wavelength, implying a raindrop is achromatic (colorless). In the model (Eq. (3)), the first term is the rain-streak term, and the second term is the background term.

3.2 Residue Channel Computation

In our method, to generate the residue channel that is free from rain streaks, we need to cancel the light chromaticity, σ , in the rain streak term in Eq. (3). For this, we employ any existing color constancy algorithm (e.g. [9]) to estimate σ , and then apply the following normalization step to the input image:

$$\mathbf{I}(\mathbf{x}) = \frac{\tilde{\mathbf{I}}(\mathbf{x})}{\sigma} = I_{rs}(\mathbf{x})\mathbf{i} + \mathbf{I}_{bg}(\mathbf{x}), \quad (4)$$

where $\mathbf{i} = (1, 1, 1)^T$, $I_{rs} = \tau\rho_{rs}L$, and $\mathbf{I}_{bg} = (T - \tau)\mathbf{B}/\sigma$. The vector division is done element wise. Note, when we normalize the image, we do not only cancel the light chromaticity, but also the color effect of the spectral sensitivities.

Therefore, based on the last equation, given a rain image \mathbf{I} , we define our residue-channel as:

$$I_{res}(\mathbf{x}) = I^M(\mathbf{x}) - I^m(\mathbf{x}), \quad (5)$$

where $I^M(\mathbf{x}) = \max\{I_r(\mathbf{x}), I_g(\mathbf{x}), I_b(\mathbf{x})\}$, and $I^m(\mathbf{x}) = \min\{I_r(\mathbf{x}), I_g(\mathbf{x}), I_b(\mathbf{x})\}$. We call I_{res} the *residue channel* of image \mathbf{I} , and it is free from rain streaks. Fig. 2 shows some examples of the residue channel. The reason why residue channel can be free from rain streaks is because the rain-streak term in Eq. (4) is achromatic, whose values are cancelled when applying Eq. (5).

To generate the residue channel, we theoretically need to apply color constancy so that the rain-streak term can be achromatic. However, in our experiments, we also noticed that even without applying color constancy, the residue channel can still work. This is because in most cases, the appearance of rain streaks is already achromatic, which is due to the dominant gray atmospheric light generated by a cloudy sky (see the discussion in the supplementary material).

3.3 Colored Residue Image

Since the residue channel is a single channel map, it has no color information. To obtain a colored residue image, we need to transform the original rain image into the YCbCr domain:

$$\begin{aligned} C_b &= 128 - (37.945/256)I_r - (74.494/256)I_g + (112.439/256)I_b, \\ C_r &= 128 - (112.439/256)I_r - (94.154/256)I_g - (18.285/256)I_b. \end{aligned} \quad (6)$$

In C_b and C_r , the achromatic value of the rain-streak term is cancelled out, since the sum of the coefficients in the definition equals to zero, and thus the two are independent



Fig. 2: **Top:** Images captured in rain scenes with rain streaks and rain accumulation. **Middle:** Residue channels of the corresponding rain images. **Bottom:** The colored residue images of the corresponding rain images. The rain streaks are significantly reduced in the residue channel and colored-residue images, though some regions become dark due to achromatic background and rain accumulation. Note, we increase the intensity of the dark regions for visualization purpose.

from rain streaks. For Y channel, however, we do not use it in our computation, since it is still affected by rain streaks. Instead, we replace it using the residue channel (from Eq. 5), which is free from rain streaks. Having obtained the values of all image intensities in the $I_{res}CbCr$ domain, we transform them back to the RGB domain, in order to obtain the colored residue image as shown in Fig. 2.

One drawback of both the residue channel and colored-residue image is that when the background is achromatic (i.e., white, gray or black), the generated background becomes dark. This is because the background term in Eq. (4) becomes achromatic, instead of a colored vector; and thus, it is cancelled out along with the rain-streak term.

3.4 Residue Image and Rain Accumulation

Rain images typically have severe rain accumulation particularly in the heavy rain. For each pixel, the intensity contributed by the rain is the accumulation of all the raindrops along the line of sight from the camera to the background object. Considering that rain accumulation is so dense that each individual streaks cannot be observed, we thus model the rain accumulation appearance similar to that of fog:

$$E(\mathbf{x}) = (1 - \alpha_t(\mathbf{x})) L + \alpha_t(\mathbf{x}) E_{bg}(\mathbf{x}), \quad (7)$$

where $\alpha_t(\mathbf{x})$ is the transmission, whose value depends on the rain droplet's attenuation factor and the distance between the camera and the background along the line of sight. This model has been successfully used in a few rain removal methods (e.g. [47,19]).

Similar to the discussion in Sec. 3.1, by taking into consideration the camera spectral sensitivities, the colored image intensity of the rain accumulation and background

can be expressed as: $\tilde{\mathbf{I}}(\mathbf{x}) = (1 - \alpha_t(\mathbf{x})) \mathbf{L} + \alpha_t(\mathbf{x}) \mathbf{B}(\mathbf{x})$. Moreover, if we incorporate the rain streak model into the rain accumulation and background, then we can express all the terms into one equation:

$$\tilde{\mathbf{I}}(\mathbf{x}) = \tau \rho_{rs}(\mathbf{x}) \mathbf{L} + (T - \tau) [(1 - \alpha_t(\mathbf{x})) \mathbf{L} + \alpha_t(\mathbf{x}) \mathbf{B}(\mathbf{x})]. \quad (8)$$

By employing a color constancy method (i.e., [9]), we can estimate the light chromaticity, σ , and use it to cancel the light chromaticity in the image:

$$\mathbf{I}(\mathbf{x}) = I_{rs}(\mathbf{x}) \mathbf{i} + I_{ra}(\mathbf{x}) \mathbf{i} + \mathbf{I}'_{bg}(\mathbf{x}), \quad (9)$$

where $I_{ra} = (T - \tau)(1 - \alpha_t)L$. $\mathbf{I}'_{bg} = \alpha_t(T - \tau)\mathbf{B}/\sigma$.

Therefore, in this all-inclusive model, Eq. (9), we have three terms: the rain-streak term (the first term), rain accumulation term (the second term), and the background term (the third term). When we see an individual streak in the input image, it is modelled by the rain streak term; however, for those rain streaks that are accumulated such that individual streaks are not distinguishable, then they are modelled by the second term.

If we apply the residue channel (Eq. (5)) or the colored-residue image to our rain model of Eq. (9), both rain streaks and rain accumulation will be removed from the images, since the rain-streak and rain-accumulation terms are both achromatic. This implies, our residue channel and colored-residue image are also free from rain accumulation. However, it comes with a price. First, for achromatic background regions, the residue channel becomes dark, since all the three terms in Eq. (9) are achromatic. Second, when rain accumulation is considerably thick, the background term is significantly suppressed, and as a result, the residue channel and image also become relatively dark, depending on the thickness of the rain accumulation. Nevertheless, in this paper, our goal is not to generate visually pleasing rain-free images, but to create an optical flow algorithm that is robust to rain. Thus, despite the presence of these dark regions, we shall see that residue channel and colored-residue image are useful tools to achieve our goal. In the subsequent section, we will discuss how to utilize the residue channel and colored-residue image to estimate optical flow robustly.

4 Decomposition Framework

In the classic variational framework, the optical flow objective function is expressed as:

$$\mathcal{L}_1(\mathbf{u}) = \sum_{\mathbf{x}} \{\Phi_D[I_1(\mathbf{x}) - I_2(\mathbf{x} + \mathbf{u})] + \lambda_s \Phi_S(\nabla \mathbf{u}(\mathbf{x}))\}, \quad (10)$$

where I_1, I_2 are the gray versions of $\mathbf{I}_1, \mathbf{I}_2$, respectively. \mathbf{u} is the flow vector with λ_s as a regularization parameter and Φ_D and Φ_S are the data and spatial penalty functions. However, as we have discussed, these I_1 and I_2 are affected by rain, and thus the BCC and GCC do not hold. The simplest idea to compute optical flow would be to use the colored-residue images as input to any modern optical flow algorithm. Unfortunately, while the colored-residue image is free from rain streaks and rain accumulation, it suffers from low contrast and the dark region effect. Hence, in our objective function, we incorporate both the colored-residue image and the input image. Our idea is that when

encountering dark regions in the residue channel, we turn to the input image for computing the flow; otherwise we use the colored-residue image. Based on this, we change our objective function:

$$\begin{aligned}\mathcal{L}_2(\mathbf{u}) = \sum_{\mathbf{x}} & \{(1 - w(\mathbf{x}))\Phi_D[I_1(\mathbf{x}) - I_2(\mathbf{x} + \mathbf{u})] \\ & + w(\mathbf{x})\Phi_D[R_1(\mathbf{x}) - R_2(\mathbf{x} + \mathbf{u})] \\ & + \lambda_s\Phi_S(\nabla\mathbf{u})\},\end{aligned}\quad (11)$$

where R_1, R_2 are the gray versions of the two colored-residue images of the two input rain frames, respectively.

Employing the input images in the objective function, however, adds some complexity. Since, besides affected by rain, raw rain images in fact have a fair amount of noise, which is not surprising since they are usually taken under dim conditions. Those who are well-versed with the art of optical flow estimation will know that this situation of low contrast and substantial noise is a sure recipe for trouble. To address the problem, our idea is to employ the structure/texture image decomposition [44] to the input images and use the structure layer extracted to provide a coarse and complementary source of flow information. While the flow information from the structure layer may be lacking in details (since the detailed textures are discarded), it is less influenced by noise. It also serves to fill in the missing information in the dark regions of the colored-residue image. Formally, the observed rain image \mathbf{I} can be modeled as a linear combination of the piecewise-smooth structure layer \mathbf{J} and the fine-detail texture layer \mathbf{K} , namely: $\mathbf{I} = \mathbf{J} + \mathbf{K}$, where the piecewise-smooth structure layer describes the principal regions of the image and the texture layer contains the fine-detailed background textures, rain streaks, and any other possible noises. The decomposition can be done by: $\min_{\mathbf{J}} \|\mathbf{I} - \mathbf{J}\|^2 + \lambda \|\nabla \mathbf{J}\|_0$, where $\nabla = (\partial x, \partial y)^T$, and λ is the weighting factor. We use the $L0$ -norm, since, being a discrete counting metric, it can deliver sharper edges and has better ability to preserve large gradients [44]. Putting all the above ideas together, we have the following unified objective function:

$$\begin{aligned}\mathcal{L}_F(\mathbf{u}, J_1, J_2) = \sum_{\mathbf{x}} & \{\lambda_d\{(1 - w(\mathbf{x}))\Phi_D[J_1(\mathbf{x}) - J_2(\mathbf{x} + \mathbf{u})] \\ & + w(\mathbf{x})\Phi_D[R_1(\mathbf{x}) - R_2(\mathbf{x} + \mathbf{u})]\} + \lambda_s\Phi_S(\nabla\mathbf{u}(\mathbf{x})) \\ & + \alpha(\|I_1(\mathbf{x}) - J_1(\mathbf{x})\|^2 + \|I_2(\mathbf{x}) - J_2(\mathbf{x})\|^2) \\ & + \beta(\|\nabla J_1(\mathbf{x})\|_0 + \|\nabla J_2(\mathbf{x})\|_0)\},\end{aligned}\quad (12)$$

where J_1, J_2 are the gray versions of the structure images of the two frames respectively.

Parameter w is the tunable weighting factor that mediates the relative contribution of the structure layer and the colored-residue image for flow computation (first and second lines of Eq. (12) respectively). It weighs more on using the colored-residue image when the residual channel is not dark. We define $w(\mathbf{x}) = \gamma I_{res}(\mathbf{x})$, where γ is a scaling factor such that there is always some contribution from the structure images. The structure-texture decomposition (third and fourth lines of Eq. (12)) is carried out jointly with the

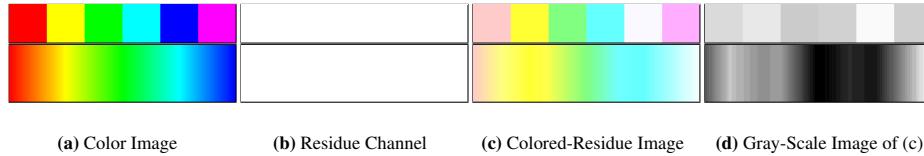


Fig. 3: **(a)** A color palette and spectrum. **(b)** The residue channel of (a). (Single channel map) **(c)** The colored-residue image of (a). **(d)** The gray scaled version of colored-residue image (c)

Algorithm 1

- ```

1: Input: Image sequence I_1, I_2 , parameters λ_s, α, β , convergence criteria ϵ
2: Initialization: Assign $J_1^{(0)} \leftarrow I_1, J_2^{(0)} \leftarrow I_2$, and initial flow $\mathbf{u}^0 \leftarrow J_1^{(0)}, J_2^{(0)}, R_1, R_2$
3: repeat (Start from i=0)
4: Compute $J_1^{(i+1)} \leftarrow \{J_2^{(i)}, I_1, \mathbf{u}^{(i)}\}, J_2^{(i+1)} \leftarrow \{J_1^{(i)}, I_2, \mathbf{u}^{(i)}\}$ (Subtask 1)
5: Estimate Flow $\{\mathbf{u}^{(i+1)} \leftarrow J_1^{(i+1)}, J_2^{(i+1)}, R_1, R_2\}$. (Subtask 2)
6: until $\|\mathbf{u}^{i+1} - \mathbf{u}^i\| < \epsilon$
7: Output: Estimated flow field $\mathbf{u}^{(M)}$

```

optical flow estimation, so that the denoising can be done in a consistent way across both frames, and that the decomposition can benefit from the redundancy of multiple frames.  $\lambda_s$  is the smoothness parameter for the flow  $\mathbf{u}$ .  $\beta$  is the parameter controlling the gradient threshold. The higher the  $\beta$ , the fewer boundaries in the piecewise-smooth background layer.  $\Phi_D$  and  $\Phi_S$  are the 'Charbonnier' penalty function for the data term and smoothness term.<sup>3</sup>

## 5 Optimization

First, we generate the residue channel maps ( $R_1, R_2$ ) from the input image and then initialize  $J_1 = I_1$ ,  $J_2 = I_2$ , and  $\mathbf{u}$  by solving Eq. (13) following the method of [7]. To optimize our objective function, we alternately solve the following subtasks until convergence:

**Subtask 1: Layer Separation** Given the current optical flow  $\mathbf{u}$ , we compute the piecewise-smooth background layer  $J_1$ , and  $J_2$  separately:

<sup>3</sup> Regarding the colored-residue images in our objective function, one may wonder the purpose of generating it, if in the end we use the gray version  $R_1, R_2$  of it. The reason is that when two objects have different colors, there are some cases where their residue channel values are identical, and thus when the objects are adjacent to each other, their color boundaries disappear, and as a result optical flow is deprived of this important information. However, if we use the gray version of the colored-residue images, we can retain the boundary information. Fig. 3 shows an example of this.

$$\begin{aligned} J_1^* &= \arg \min_{J_1} \sum_{\mathbf{x}} \{ \lambda_d \Phi_D [J_1(\mathbf{x}) - J_2(\mathbf{x} + \mathbf{u})] + \alpha |I_1(\mathbf{x}) - J_1(\mathbf{x})|^2 + \beta |\nabla J_1(\mathbf{x})|_0 \} \\ J_2^* &= \arg \min_{J_2} \sum_{\mathbf{x}} \{ \lambda_d \Phi_D [J_1(\mathbf{x}) - J_2(\mathbf{x} + \mathbf{u})] + \alpha |I_2(\mathbf{x}) - J_2(\mathbf{x})|^2 + \beta |\nabla J_2(\mathbf{x})|_0 \} \end{aligned}$$

The objective functions are not convex due to the  $L_0$ -norm terms, unlike the standard structure-texture decomposition. To resolve this problem, we adopt the alternating optimization strategy from [44], by introducing two auxiliary variables to decouple the unsmooth gradient term and the smooth quadratic terms.

**Subtask 2: Optical Flow Computation** Given current piecewise-smooth background layers ( $J_1, J_2$ ), we estimate the optical flow vector  $\mathbf{u}$  following the method of [7]:

$$\begin{aligned} \mathbf{u}^* = \arg \min_{\mathbf{u}} \sum_{\mathbf{x}} \{ & \lambda_d ((1 - w(\mathbf{x})) \Phi_D [J_1(\mathbf{x}) - J_2(\mathbf{x} + \mathbf{u})] \\ & + w(\mathbf{x}) \Phi_D [R_1(\mathbf{x}) - R_2(\mathbf{x} + \mathbf{u})]) + \lambda_s \Phi_S (\nabla \mathbf{u}(\mathbf{x})) \} \end{aligned} \quad (13)$$

Although there is no guarantee for convergence to this non-convex problem, with initialization as proposed above, this algorithm performs well in practice. In our experiments, we run our algorithm on hundreds of different rain scenes and it showed good convergence. A video of is attached in supplementary material to demonstrate the stability, robustness and the convergence of the proposed method.

## 6 Experiments

**Ablation Study** To study how the colored-residue image and structure layer complement each other, we conduct the following ablation experiments using a dataset [27] rendered with rain. We compare the performance of our algorithm with a few baseline methods.

First, we evaluate the performance of the colored-residue image alone by subjecting it to increasingly dense rain streaks and increasing levels of rain accumulation along with the additive white Gaussian noise (Fig. 4 the top two rows). The optical flow results are shown on the top row of Fig. 4 (a)(b). As can be seen, while the colored-residue image is effective in dealing with rain streaks, it is negatively affected by the low contrast brought by the rain accumulation. This is where the structure layer (or the decomposition) comes in. Second, we evaluate the performance of the structure layer alone similarly by subjecting it to increasingly dense rain streaks and increasing levels of rain accumulation. From the result in the top row of Fig. 4 (c),, the structure alone does not achieve good performance, since rain streaks in the original input can have rather strong gradients and remain in the structure layer. However, when combined the colored-residue image and structure layer, the performance improves, showing a graceful degradation of performance as rain increases (also in Fig. 4 (c)). More results of our ablation study including on real images are available in the supplementary material. The bottom row of Fig. 4 shows the qualitative results of using the colored-residue only, the structure layer only, and our combination.

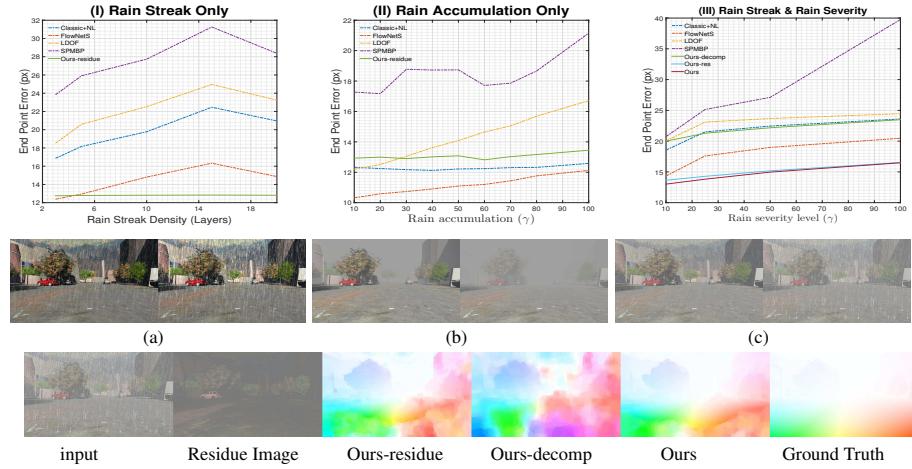


Fig. 4: Ablation study. Top two rows: Input images and the performance graph of the proposed method on the driving dataset rendered with (a) rain streaks alone, (b) rain accumulation alone, (c) combining rain streaks and rain accumulation. Bottom row: Input and optical flow qualitative results using the colored-residue image alone (ours-residue), structure layer alone (ours-decomp), and combined (ours).

**Evaluation Datasets** To obtain optical flow ground-truths for real images is considerably difficult, however it is even more difficult for rain scenes. Baker et al. [2] obtain ground-truth data of only a couple of real image pairs using a controlled experiment setup, which does not work under outdoor rain. Using a LIDAR system to obtain flow ground truths is also problematic, since layers of densely accumulated raindrops will absorb and reflect laser rays, which can lead to missing data points and wrong measurements in the echo-backed results. Hence, in this paper, for quantitative evaluations, we use a few different strategies. First, we generate synthetic rain by following the rain model [15] on Middlebury [2], Sintel[8] and KITTI [24] optical flow datasets. Second, we combine real rain images with synthesized object motions, creating a new hybrid dataset named FVR-660, which the ground-truths are known. There are in total 660 sequences in this dataset. The top row of Fig. 5 shows some examples. Third, we introduce our NUS-100 dataset containing 100 sequences of real rain and real motion, whose ground truth is obtained by human annotation. An example is shown in the bottom row of Fig. 5. The details of FVR-660 and NUS-100 dataset generation are included in the supplementary material.<sup>4</sup>

**Synthetic Rain Results** Using our synthetic data, we compare our algorithm with a few conventional methods, i.e. Classic+NL [34], LDOF [6], and SP-MBP [18], EpicFlow [31], as well as recent deep learning methods such as FlowNet2 [17], DCFlow [42] and FlowNet [11], specifically the FlowNetS variant. For a fair comparison, we utilize the

<sup>4</sup> FVR-600 and NUS-100 datasets are available at : <https://liruoteng.github.io/RobustOpticalFlowProject>



Fig. 5: **Top:** Two examples from the Flying Vehicles with Rain (FVR-660). From left to right are the generated image pair, and the color coded flow field ground truth. **Bottom:** An example of NUS-100 dataset. From left to right are the input image pair, the annotated labels for objects with motion, the horizontal component of the flow, and the flow ground truth.

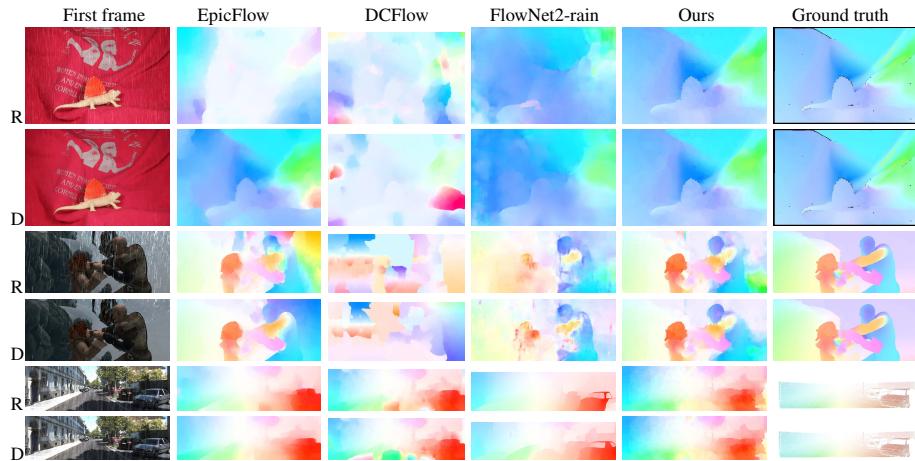


Fig. 6: Method comparisons on Middlebury, MPI Sintel, and KITTI datasets, which are all rendered with rain. The first column "R" and "D" represent synthesized rain sequences and the same sequences after [47]'s deraining method. (**Best zoomed in on screen**).

recent deraining method [47] as a preprocessing step for these methods. The quantitative results are shown in columns 1 to 3 of Table 1. The qualitative results of these comparisons are shown in Fig. 6. In the figure, the original synthesized rain image is denoted with 'R', and the image produced by the deraining operation is denoted with 'D'. FlowNet2 [17] and FlowNetS [11] are not originally trained using rain images, and thus may not perform well under rain conditions. Hence, we render the Flying Chair dataset [11] with synthetic rain streaks using the same rain streak model as the test dataset. We then fine-tune FlowNetS and FlowNet2 end-to-end on this dataset and pick the best performed model for evaluation. The fine-tuned models are denoted as FlowNetS-rain and FlowNet2-rain respectively.

Table 1: A comparison of our algorithm with several top-performing methods on synthesized rain datasets. "De-rain" indicates the results of each method performed on the sequences after Yang et al.'s [47] de-rain method.

| Method               | Middlebury  |              |             | Sintel      |             |             | KITTI2012   |             |             | FVR-660     |             | NUS-100     |             |
|----------------------|-------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                      | Rain        | De-rain      | Clean       | Rain        | De-rain     | Clean       | Rain        | De-rain     | Clean       | Rain        | De-rain     | Rain        | De-rain     |
| Classic+NL-fast [34] | 0.90        | 0.60         | <b>0.22</b> | 7.97        | 6.79        | 4.94        | 9.17        | 9.14        | 3.84        | 2.17        | 2.19        | 0.49        | 0.53        |
| LDOF [6]             | 0.90        | 0.66         | 0.45        | 10.68       | 6.70        | 4.29        | 10.17       | 9.90        | 5.05        | 2.93        | 2.98        | 0.68        | 0.60        |
| SP-MBP [18]          | 0.93        | 0.60         | 1.92        | 12.32       | 7.06        | 3.56        | 15.71       | 15.94       | 6.01        | 5.37        | 5.50        | 0.50        | 0.55        |
| EpicFlow [31]        | 1.57        | 0.48         | 0.35        | 14.92       | 7.03        | 2.46        | 6.94        | 6.78        | 1.44        | 4.52        | 4.44        | 0.35        | 0.36        |
| DCFFlow [42]         | 1.62        | 0.68         | 0.35        | 10.68       | 6.70        | 2.28        | 10.17       | 6.70        | <b>1.04</b> | 46.71       | 30.69       | 0.30        | 0.30        |
| FlowNetS [11]        | 2.58        | 1.54         | 1.09        | 7.40        | 6.85        | 4.50        | 17.43       | 18.73       | 4.27        | 2.18        | 2.55        | 0.53        | 0.55        |
| FlowNetS-Rain        | 1.45        | 1.42         | 1.28        | 6.90        | 5.82        | 4.65        | 6.84        | 6.91        | 5.68        | 2.85        | 3.79        | 0.63        | 0.65        |
| FlowNet2 [17]        | 0.79        | 0.58         | 0.40        | 7.68        | 8.56        | <b>2.13</b> | 7.23        | 8.64        | 2.22        | 5.73        | 6.07        | 0.28        | 0.30        |
| FlowNet2-Rain        | 1.24        | 1.11         | 0.32        | 7.54        | 6.30        | 5.22        | 8.01        | 8.13        | 6.70        | 2.21        | 2.18        | 0.42        | 0.43        |
| Ours                 | <b>0.30</b> | <b>0.332</b> | 0.32        | <b>5.46</b> | <b>5.06</b> | 4.71        | <b>6.65</b> | <b>6.67</b> | 4.05        | <b>1.76</b> | <b>1.81</b> | <b>0.22</b> | <b>0.19</b> |

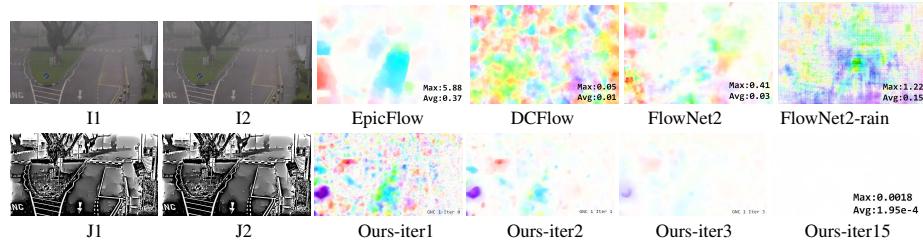


Fig. 7: Static Scene Analysis. **I1, I2** are captured rainy image pair of a static scene. **J1, J2** are the corresponding piecewise-smooth layers of **I1, I2** respectively. Flow map produced by competing algorithms are shown in top row. Flow map produced by our method at different optimization stages are shown in bottom row.

**Real Rain Results** To verify the effectiveness of our algorithm, we perform a sanity check on the estimated flow for static real-rain image pairs as shown in Fig. 7. Since this is a static scene under heavy rain, the true optical flow for the background should be zero everywhere. From the figure (the top row), one can see that the baseline methods produce erroneous flow due to the motion of the rain. In comparison, the result of our algorithm shows a significantly cleaner result. The average magnitude of our flow field is 0.000195 pixel, which is essentially zero flow. Moreover, the plots in the bottom row also show that during the iteration process, the optical flow estimation improves. This means that the structure layer does provide complementary information to the colored-residue image.

We also compare the baseline methods with our algorithm on the FVR-660 dataset for quantitative evaluation (column 5 of Table 1) and qualitative evaluation (Fig. 8). For this evaluation, the deraining preprocessing [47] is applied to the existing methods. As one can see from Fig. 8, the results of the baseline methods contain obvious erroneous flow due to the presence of rain streaks. The state-of-the-art deraining method does not generalize well on different rain types, hence rain streaks are not removed clearly and some deraining artifacts may also be introduced. Finally, we compare our algorithm

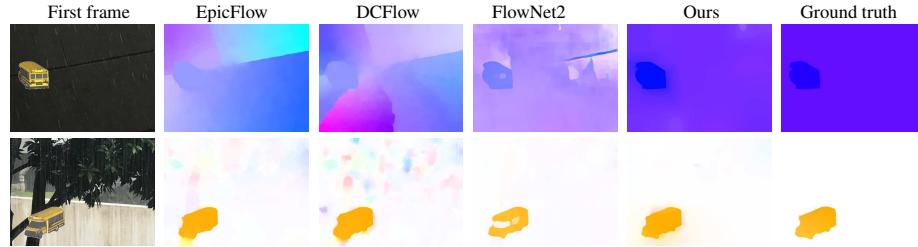


Fig. 8: Method comparison on Flying Vehicle with Rain (FVR-660) dataset. (**Best viewed on screen**).

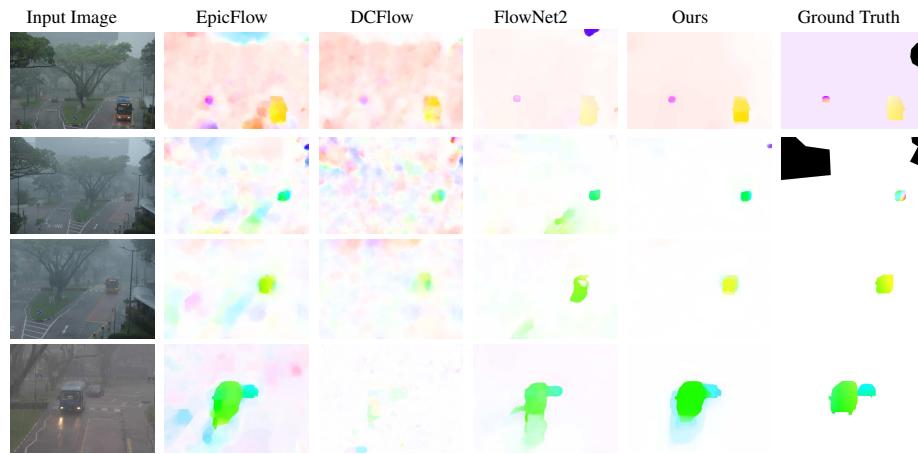


Fig. 9: Method comparison on real rainy scenes with different severity level. The last column is annotated ground truth using [20]. The black region in Ground Truth indicates invalid region, which is not counted in flow evaluation. (**Best zoomed in on screen**).

with baseline methods on the manually annotated real rainy sequences in the NUS-100 dataset. The quantitative result is included in column 6 of Table 1 and qualitative results are shown in Fig. 9.

## 7 Conclusion

We have introduced a robust algorithm for optical flow in rainy scenes. To our knowledge, it is the first time an optical flow algorithm is specifically designed to deal with rain. Through this work, we make a few contributions. We introduced the residue channel and colored-residue image that are both free from rain streaks and rain accumulation. We proposed an integrated framework to deal with rain that combine the residue channel, colored-residue image, and piecewise-smooth structure layer extraction. We provide a rain optical flow benchmark containing both synthesized motion and real motion to the public.

## References

1. Bailer, C., Taetz, B., Stricker, D.: Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. CoRR **abs/1508.05151** (2015), <http://arxiv.org/abs/1508.05151>
2. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. International Journal of Computer Vision **92**(1), 1–31 (Mar 2011)
3. Barnum, P., Kanade, T., Narasimhan, S.: 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. pp. 8–p. INRIA (2007)
4. Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. Int. J. Comput. Vision **12**(1), 43–77 (Feb 1994). <https://doi.org/10.1007/BF01420984>, <http://dx.doi.org/10.1007/BF01420984>
5. Black, M.J., Anandan, P.: The robust estimation of multiple motions. Comput. Vis. Image Underst. **63**(1), 75–104 (Jan 1996). <https://doi.org/10.1006/cviu.1996.0006>
6. Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(3), 500–513 (2011), <http://lmb.informatik.uni-freiburg.de/Publications/2011/Bro11a>
7. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. International Journal of Computer Vision **61**(3), 211–231 (Feb 2005). <https://doi.org/10.1023/B:VISI.0000045324.43199.43>, <https://doi.org/10.1023/B:VISI.0000045324.43199.43>
8. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A Naturalistic Open Source Movie for Optical Flow Evaluation, pp. 611–625. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
9. Cheng, D., Prasad, D.K., Brown, M.S.: Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. JOSA A **31**(5), 1049–1058 (2014)
10. Choy, C.B., Gwak, J., Savarese, S., Chandraker, M.: Universal correspondence network. In: Advances in Neural Information Processing Systems 29 (2016)
11. Dosovitskiy, A., Fischer, P., Ilg, E., Golkov, V., Husser, P., Hazrba, C., Golkov, V., Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: IEEE International Conference on Computer Vision (ICCV) (2015)
12. Garg, K., Nayar, S.K.: Vision and rain. Int. J. Comput. Vision
13. Garg, K., Nayar, S.K.: Photometric model of a rain drop. Tech. rep. (2003)
14. Garg, K., Nayar, S.K.: Detection and removal of rain from videos. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. vol. 1, pp. I–I. IEEE (2004)
15. Garg, K., Nayar, S.K.: Photorealistic rendering of rain streaks. ACM Trans. Graph. **25**(3), 996–1002 (Jul 2006). <https://doi.org/10.1145/1141911.1141985>
16. Horn, B.K.P., Schunck, B.G.: Determining optical flow. ARTIFICAL INTELLIGENCE **17**, 185–203 (1981)
17. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. CoRR **abs/1612.01925** (2016), <http://arxiv.org/abs/1612.01925>
18. Li, Y., Min, D., Brown, M.S., Do, M.N., Lu, J.: Spm-bp: Sped-up patchmatch belief propagation for continuous mrfs. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 4006–4014 (Dec 2015). <https://doi.org/10.1109/ICCV.2015.456>

19. Li, Y., Tan, R.T., Guo, X., Lu, J., Brown, M.S.: Rain streak removal using layer priors. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
20. Liu, C., Freeman, W.T., Adelson, E.H., Weiss, Y.: Human-assisted motion annotation. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (June 2008). <https://doi.org/10.1109/CVPR.2008.4587845>
21. Liu, C., Yuen, J., Torralba, A.: Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(5), 978–994 (May 2011). <https://doi.org/10.1109/TPAMI.2010.147>, <http://dx.doi.org/10.1109/TPAMI.2010.147>
22. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2. pp. 674–679. IJCAI’81, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1981), <http://dl.acm.org/citation.cfm?id=1623264.1623280>
23. Luo, Y., Xu, Y., Ji, H.: Removing rain from a single image via discriminative sparse coding. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 3397–3405 (Dec 2015). <https://doi.org/10.1109/ICCV.2015.388>
24. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
25. Mileva, Y., Bruhn, A., Weickert, J.: Illumination-Robust Variational Optical Flow with Photometric Invariants, pp. 152–162. Springer Berlin Heidelberg, Berlin, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-74936-3\\_16](https://doi.org/10.1007/978-3-540-74936-3_16), [http://dx.doi.org/10.1007/978-3-540-74936-3\\_16](http://dx.doi.org/10.1007/978-3-540-74936-3_16)
26. Mohamed, M.A., Rashwan, H.A., Mertsching, B., Garca, M.A., Puig, D.: Illumination-robust optical flow using a local directional pattern. *IEEE Transactions on Circuits and Systems for Video Technology* **24**(9), 1499–1508 (Sept 2014). <https://doi.org/10.1109/TCSVT.2014.2308628>
27. N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, 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), <http://lmb.informatik.uni-freiburg.de/Publications/2016/MIFDB16>, arXiv:1512.02134
28. R. Richter, S., Hayder, Z., Koltun, V.: Playing for benchmarks (09 2017)
29. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. CoRR **abs/1611.00850** (2016), <http://arxiv.org/abs/1611.00850>
30. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. CoRR **abs/1611.00850** (2016), <http://arxiv.org/abs/1611.00850>
31. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In: Computer Vision and Pattern Recognition (2015)
32. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Deepmatching: Hierarchical deformable dense matching. *International Journal of Computer Vision* **120**(3), 300–323 (Dec 2016)
33. Scharstein, D., Szeliski, R., Zabih, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In: Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001). pp. 131–140 (2001). <https://doi.org/10.1109/SMBV.2001.988771>
34. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 2432–2439. IEEE (Jun 2010)
35. Szeliski, R., Avidan, S., Anandan, P.: Layer extraction from multiple images containing reflections and transparency. In: Proceedings IEEE Conference on Computer Vision and

- Pattern Recognition. CVPR 2000 (Cat. No.PR00662). vol. 1, pp. 246–253 vol.1 (2000). <https://doi.org/10.1109/CVPR.2000.855826>
36. Szeliski, R.: Image alignment and stitching: A tutorial. *Found. Trends. Comput. Graph. Vis.* **2**(1), 1–104 (Jan 2006). <https://doi.org/10.1561/0600000009>, <http://dx.doi.org/10.1561/0600000009>
  37. Trobin, W., Pock, T., Cremers, D., Bischof, H.: An unbiased second-order prior for high-accuracy motion estimation. In: Pattern Recognition, 30th DAGM Symposium, Munich, Germany, June 10–13, 2008, Proceedings. pp. 396–405 (2008). [https://doi.org/10.1007/978-3-540-69321-5\\_40](https://doi.org/10.1007/978-3-540-69321-5_40), [http://dx.doi.org/10.1007/978-3-540-69321-5\\_40](http://dx.doi.org/10.1007/978-3-540-69321-5_40)
  38. Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D.: An improved algorithm for tv-l1 optical flow. In: Cremers, D., Rosenhahn, B., Yuille, A.L., Schmidt, F.R. (eds.) *Statistical and Geometrical Approaches to Visual Motion Analysis*. pp. 23–45. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
  39. Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D.: Statistical and geometrical approaches to visual motion analysis. chap. An Improved Algorithm for TV-L1 Optical Flow, pp. 23–45. Springer-Verlag, Berlin, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-03061-1\\_2](https://doi.org/10.1007/978-3-642-03061-1_2), [http://dx.doi.org/10.1007/978-3-642-03061-1\\_2](http://dx.doi.org/10.1007/978-3-642-03061-1_2)
  40. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: DeepFlow: Large displacement optical flow with deep matching. In: IEEE International Conference on Computer Vision (ICCV). Sydney, Australia (Dec 2013), <http://hal.inria.fr/hal-00873592>
  41. Xiao, J., Cheng, H., Sawhney, H., Rao, C., Isnardi, M.: Bilateral filtering-based optical flow estimation with occlusion detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *Computer Vision – ECCV 2006*. pp. 211–224. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
  42. Xu, J., Ranftl, R., Koltun, V.: Accurate Optical Flow via Direct Cost Volume Processing. In: CVPR (2017)
  43. Xu, J., Ranftl, R., Koltun, V.: Accurate optical flow via direct cost volume processing. CoRR **abs/1704.07325** (2017), <http://arxiv.org/abs/1704.07325>
  44. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via l0 gradient minimization. ACM Transactions on Graphics (SIGGRAPH Asia) (2011)
  45. Yang, H., Lin, W.Y., Lu, J.: Daisy filter flow: A generalized discrete approach to dense correspondences. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3406–3413 (June 2014). <https://doi.org/10.1109/CVPR.2014.435>
  46. Yang, J., Li, H., Dai, Y., Tan, R.T.: Robust optical flow estimation of double-layer images under transparency or reflection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1410–1419 (June 2016). <https://doi.org/10.1109/CVPR.2016.157>
  47. Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z., Yan, S.: Joint rain detection and removal via iterative region dependent multi-task learning. CoRR **abs/1609.07769** (2016), <http://arxiv.org/abs/1609.07769>