

Supplementary Material for 'Robust Optical Flow in Rainy Scenes'

Ruoteng Li¹[0000-1111-2222-3333], Robby T. Tan^{1,2}[1111-2222-3333-4444], and
Loong-Fah Cheong¹[2222--3333-4444-5555]

¹ National University of Singapore

² Yale-NUS College

1 Rain Streak Image Formation

In the main paper Eq.(2), we have explicitly express the image intensity with rain streaks as:

$$\tilde{\mathbf{I}}(\mathbf{x}) = \tau \int_{\Omega} \bar{E}_{rs}(\mathbf{x}, \lambda) q_c(\lambda) d\lambda + (T - \tau) \int_{\Omega} E_{bg}(\mathbf{x}, \lambda) q_c(\lambda) d\lambda, \quad (1)$$

Considering the model in Eq. 1, to be more explicit with the presence of the color and intensity of the atmospheric light in the model, we define $E_{rs}(\mathbf{x}, \lambda) = \rho_{rs}(\mathbf{x})L(\lambda)$ and $E_{bg}(\mathbf{x}, \lambda) = \rho_{bg}(\mathbf{x}, \lambda)L(\lambda)$, where L is the atmospheric light, ρ_{rs} is composed of refraction, specular reflection, and internal reflection coefficients of a raindrop [6], and ρ_{bg} is the reflectance of the background object. We assume that ρ_{rs} is independent from wavelength, implying raindrop is achromatic (colorless). We also assume that $L(\lambda)$, which is dominated by the atmospheric light, is uniform across the scene. Focusing on each of the terms of Eq. (1), we can write:

$$\tilde{\mathbf{I}}_{rs}(\mathbf{x}) = \tau \rho_{rs}(\mathbf{x}) \int_{\Omega} L(\lambda) q_c(\lambda) d\lambda = \tau \rho_{rs}(\mathbf{x}) \mathbf{L}, \quad (2)$$

$$\tilde{\mathbf{I}}_{bg}(\mathbf{x}) = (T - \tau) \int_{\Omega} \rho_{bg}(\mathbf{x}, \lambda) L(\lambda) q_c(\lambda) d\lambda = (T - \tau) \mathbf{B}, \quad (3)$$

where $\mathbf{L} = (L_r, L_g, L_b)^T$, which represents the light color vector, and $\mathbf{B} = (B_r, B_g, B_b)^T$ which represents the background color vector. Thus, $\tilde{\mathbf{I}}(\mathbf{x}) = \tilde{\mathbf{I}}_{rs}(\mathbf{x}) + \tilde{\mathbf{I}}_{bg}(\mathbf{x})$. Furthermore, using chromaticity, we can express the model as:

$$\tilde{\mathbf{I}}(\mathbf{x}) = \tau \rho_{rs}(\mathbf{x}) L \boldsymbol{\sigma} + (T - \tau) B \boldsymbol{\pi}, \quad (4)$$

where $\boldsymbol{\sigma} = (\sigma_r, \sigma_g, \sigma_b)^T$, $\boldsymbol{\pi} = (\pi_r, \pi_g, \pi_b)^T$ are the chromacities of \mathbf{L} and \mathbf{B} , respectively, $L = L_r + L_b + L_g$ and $B = B_r + B_b + B_g$. We define $\boldsymbol{\sigma} = \mathbf{L}/L$ and $\boldsymbol{\pi} = \mathbf{B}/B$. L and B represent, respectively, the brightness values of the light and reflection of the background objects. Both of them are scalar values. Hence, we have obtained the rain streak image formation as shown in Eq.(3) of the main paper.

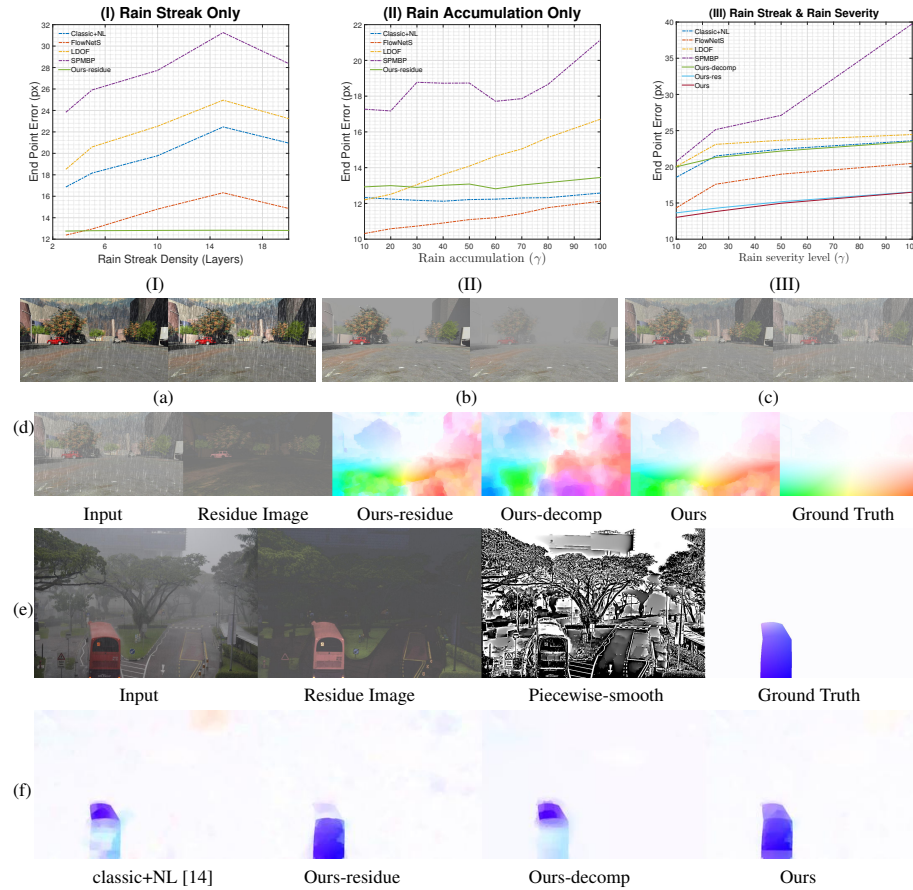


Fig. 1: **(I, II, III)** are the performance graphs of our method and baseline methods on the synthesized driving dataset (a,b,c) respectively. **(a)**: rain streaks only. **(b)**: Rain accumulation effect only. **(c)**: Both rain streaks and rain accumulation. **(d-f)**: Comparison between our method and baseline methods on synthesized driving dataset and real rain sequence.

2 Ablation Study Details

In this supplementary material, we discuss our ablation study further. We denote our algorithm with colored residue image component alone as **Ours-residue**, and with decomposition component alone as **Ours-decomp**. The full version of our method is denoted as **Ours**.

Colored Residue Image vs. Rain Streak Density As mentioned in our main paper, the colored residue image is independent from rain streaks, hence computing optical flow based on it should be more robust compared to that of using the original input image. We

rendered photorealistic rain streaks [7] with gradually increased density on the driving sequences [12] (shown in 1(a)). We represent the density levels by the number of layers, with more layers leading to a denser accumulation of rain streaks. The performance of our algorithm and baseline methods are shown in 1(I). As can be seen the performance of Our-residue is not affected too much by the rain streaks density. This experiment shows that our algorithm is quite robust under rain streaks only scenario when using colored residue image component alone.

Colored Residue Image vs. Rain Accumulation The performance of colored residue image only may be degraded when there is strong fog-like rain accumulation, which causes the scene to be low contrast and in turn makes the colored residue image produce dark regions. To study how the rain accumulation affects the performance of colored residue image, we gradually increase synthesized fog-like rain accumulation with additive white Gaussian noise (sigma = 0.2) on the sequence (shown in 1(b)). Using the fog model: $\mathbf{I}(\mathbf{x}) = \alpha(\mathbf{x})\mathbf{B}(\mathbf{x}) + (1 - \alpha(\mathbf{x}))\mathbf{L}(\mathbf{x})$, where $\alpha(\mathbf{x}) = \exp(-\gamma d(\mathbf{x}))$, L is the atmospheric light and \mathbf{B} is the radiance of the background, $\alpha(\mathbf{x})$ is the transmittance and $d(\mathbf{x})$ is the distance of the object at point \mathbf{x} . Employing the model, we adjust $\gamma \in [10, 100]$ to control the thickness of the rain accumulation effect (veiling effect). The results are shown in 1(II).

Decomposition vs. Rain Severity To deal with loss of contrast due to heavy rain and accumulation, we propose the use of the structure and texture layers decomposition (Sect 3.2). We render gradually increased dense rain streaks [7] and noisy rain accumulation together on the sequences (shown in Fig. 1(c)). As can be seen from the results in Fig. 1(III), **our-decomp** alone does not show improvement, since some rain streaks might be considered as part of the structure layer that is used in the optical flow computation. On the contrary, combining the colored-residue image and structure layer does help the optical flow computation to be more robust under dense rain streaks and heavy rain accumulation scenarios.

Fig. 1 (d)-(f) demonstrates the results of running our method with each component inactivated on both synthesized and real rain sequences. As can be seen, the decomposition help **Ours-residue** improve over the regions (sky, near field road, remote tree) with low contrast (compare Fig. 1(d, Ours-residue and Ours)), whereas residue image helps **Ours-decomp** to filter out the effect of rain streaks in front of the bus and the raindrop splash around the bus top area (compare Fig. 1(f, Ours-decomp and Ours)). One may want to zoom in Fig. 1(e, Residue Image) to see that there is no rain streak left on the bus. On the other hand, in (e) piecewise-smooth image, we can see the remote objects clearly, although in the original rain image they suffer from low contrast issue due to the heavy rain accumulation.

3 Details of FVR-660 Dataset Generation

Inspired by the Flying Chair dataset [5], we create a real rain dataset that can quantitatively evaluate our method. In generating the Flying Vehicle with Rain (FVR) dataset,



Fig. 2: Examples of vehicle models used in FVR-660 dataset. **Top:** From left to right are example of an image pair, and the vehicle models used to render on background image. **Bottom:** From left to right are the rain background rendered with vehicles, and the vehicles without/with applying rain streaks and rain accumulation.

we use 14 pairs of image of resolution 4032 x 3024 pixels as background. For foreground objects, we use 24 vehicle models (3D) from TrimbleTM, from which we capture 30 views: 15 azimuth angles (α) and 2 elevation angles (ψ). Hence, we create a vehicle image set containing 720 samples. Examples of the vehicle models of different views are shown in Fig. 2. In the FVR-660 dataset, to generate the first image of an image pair, we randomly position (with uniform distribution) in the images multiple vehicles randomly selected from the vehicle image set. The size of the vehicles are sampled from a Gaussian with mean of 200 (pixel) and standard deviation 200, but with the vehicle size limited in [50, 400] pixels. An example of the first image of an image pair is shown in Fig 2(a).

To generate the second image in an image pair and the flow field, we apply 2D Euclidean transformation independently on the vehicles and the background image. The parameters of the translation vector (T_x, T_y) and rotation angle (θ) are sampled from a family of a power of Gaussian following [5]. Let $\gamma \sim \mathcal{N}(\mu, \sigma)$ as a univariate Gaussian parameter, the composed formula is expressed in Eq. (5).

$$\xi = \max(\min(\text{sign}(\gamma)|\gamma|^k, b), a), \quad (5)$$

We denote the distribution of ξ by $G(k, \mu, \sigma, a, b)$ and all transformation parameters are sampled from this distribution with the parameters shown in Table 1.

After generating the second image and flow field, we will render the rain streaks and rain accumulation on top of the vehicles. Here, we adopt state-of-the-art methods [16] and [4] to extract rain streaks and fog-like rain accumulation effect respectively from the background image and add them back on top of the vehicles. The rain streak and fog rendered images are shown in Fig. 2. Then we cut each synthesized image

Parameter	k	μ	σ	a	b
Background Translation T_{bx}	4	0	2.3	-15	15
Background Translation T_{by}	4	0	2.3	-15	15
Vehicle Translation T_{vx}	3	0	2.3	-5	5
Vehicle Translation T_{vy}	3	0	2.3	-5	5
Vehicle Translation θ_v	2	0	2.3	-5	5

Table 1: The distribution of the transformation parameters.

pair into 49 smaller images of resolution 512×385 pixels each, and obtain 686 image pairs in total. We manually remove those resultant images which only contains sky as background because there is no texture on background image. As a result, this dataset contains 660 real rain image pairs with optical flow ground truth. It is meant for algorithm testing and evaluation in our experiments. Since this dataset is inspired by the FlyingChair dataset [5], but we use vehicles instead of chairs, we call it 'Flying Vehicles with Rain' (FVR) dataset. However, similar to FlyingChairs, this dataset only contains 2D Euclidean motion, which may not fully represent real-world motions.

4 Details of X-100 Dataset Generation

This dataset contains 100 real rain sequences of different rain types and rain severity levels, including heavy rain accumulation and rain streaks. In these sequences, we manually delineated all moving objects (cars, trucks, buses, pedestrians, tree leaves, etc.) in the scenes using the labeling tool provided by [11]. Among these objects, we only estimated the motion ground truth for rigid bodies such as cars, buses and the scene background. Other moving objects are marked as 'invalid' and are excluded in the optical flow evaluation. To obtain reliable motion ground truths, we choose the semi-automatic motion labelling scheme from [11] and select the 'affine mode' to compute the motion of the moving objects, given that these objects span a small field of view. The details of creating this dataset can be found in the supplementary material. While this dataset has some limitations, e.g., non-rigid body motion is ignored and human annotation may come with errors, it offers the best ground truths that we can possibly obtain for real rain and real motion.

5 Verification of Residue Channel Prior

To verify how good the residue channel is to reduce rain streaks for improved optical flow estimation, we collect a set of outdoor images of rainy scenes and manually select more than 514 images containing obvious rain streaks (shown in Fig. 3(a)). We crop out rain streak regions with uniform textureless background from the selected rain images and resize them to patches of size $150\text{px} \times 150\text{px}$. (Fig. 3(b)). The corresponding residue channels of these images and patches are computed as shown at the bottom of Fig. 3. We then compute the variance of each image patch and the variance of its corresponding residue channel. Fig. 3(d) demonstrates the variance histogram over all

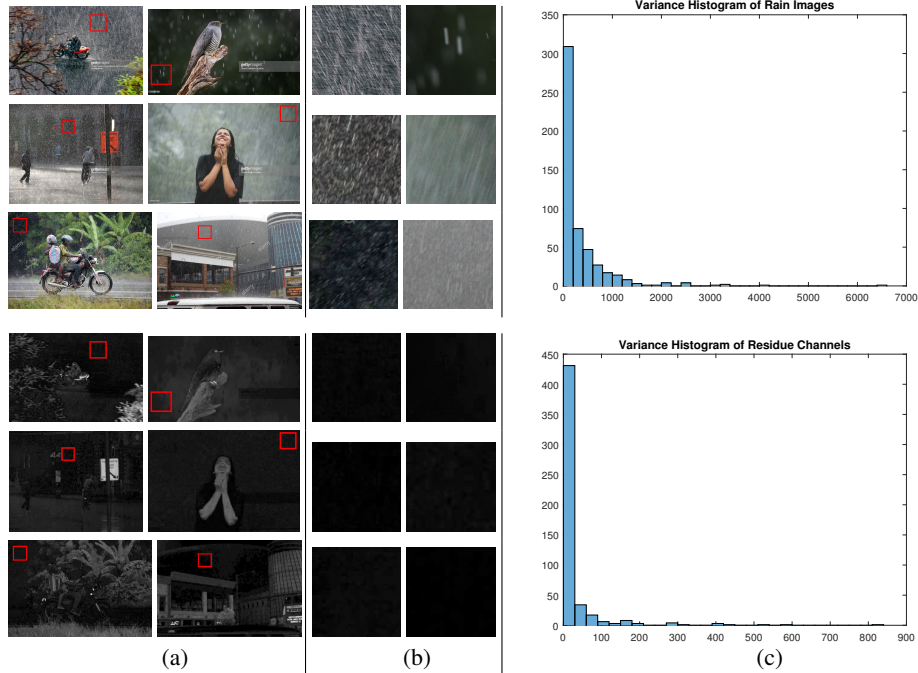


Fig. 3: (a) Collected rain images (top) and their corresponding residue channels (bottom) (b) Cropped patches of regions with uniform background (top) and their corresponding residue channels (bottom). (c) Top: Variance Histogram of the intensity of the pixels in all 514 cropped patches. Bottom: Variance Histogram of the intensity of pixels the corresponding 514 residue channels.

514 patches and their corresponding residue channels. Since the background of each cropped patch is uniform and textureless, the intensity of its pixels should remain almost uniform, meaning the variance of the non-rain background should be very small. As one can see that the patches' residue channels are really dark and the variance is significantly reduced. 88.91% of the residue patches have variance smaller than 50. However, only 36.77% rain image patches have variance less than 50. This is a strong support to our hypothesis that residue channel, even without color constancy step, is able to significantly reduce the effect of rain streaks in optical flow estimation.

6 Algorithm Processing Speed

Table 2 shows the processing speed of our algorithm and competing methods. All the methods are tested on an image pair of MPI Sintel dataset on a PC with Intel Core i7-6850K CPU and NVIDIA GTX1080Ti GPU. As our method is developed based on Horn-Schunck framework, the processing speed of our method is limited by the required iterative optimization and large matrix division. The residue image computation and the piecewise fine-detail decomposition only added marginal computation load. In

Table 2: A computation efficiency comparison of our algorithm with several top-performing methods.

Method	Run Time (s)	Platform
DCFlow[15]	3.83	GPU
FlowNetS [5]	0.08	GPU
FlowNetS-Rain	0.08	GPU
FlowNet2 [9]	0.10	GPU
FlowNet2-Rain	0.10	GPU
SP-MBP [10]	34.83	C++
EpicFlow [13]	18.53	C++
Classic+NL-fast [14]	47.51	MATLAB
Classic+NL [14]	255.30	MATLAB
LDOF [1]	73.02	MATLAB
MirrorFlow [8]	1913.05	MATLAB
Ours	64.94	MATLAB

addition, a series of real-time implementation of this algorithm is available in the literature. [2][3]. The proposed method is able to be improved to a real-time implementation as a future work.

7 Video

We enclosed two video files in this supplementary material package. One video demonstrates the result of our algorithm performed on a couple of real rain long sequences. The other compares our algorithm with a few baseline methods.

References

1. 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>
2. Bruhn, A., Weickert, J., Feddern, C., Kohlberger, T., Schnorr, C.: Variational optical flow computation in real time. *IEEE Transactions on Image Processing* **14**(5), 608–615 (May 2005). <https://doi.org/10.1109/TIP.2005.846018>
3. Bruhn, A., Weickert, J., Feddern, C., Kohlberger, T., Schnorr, C.: Real-time optic flow computation with variational methods. In: Petkov, N., Westenberg, M.A. (eds.) *Computer Analysis of Images and Patterns*. pp. 222–229. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
4. Cai, B., Xu, X., Jia, K., Qing, C., Tao, D.: Dehazenet: An end-to-end system for single image haze removal. *IEEE Trans. Image Processing* **25**(11), 5187–5198 (2016). <https://doi.org/10.1109/TIP.2016.2598681>, <https://doi.org/10.1109/TIP.2016.2598681>
5. 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)
6. Garg, K., Nayar, S.K.: Photometric model of a rain drop. Tech. rep. (2003)
7. 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>
8. Hur, J., Roth, S.: Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation. In: *The IEEE International Conference on Computer Vision (ICCV)* (Oct 2017)
9. 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>
10. 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>
11. 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>
12. 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
13. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In: *Computer Vision and Pattern Recognition* (2015)
14. 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)
15. Xu, J., Ranftl, R., Koltun, V.: Accurate Optical Flow via Direct Cost Volume Processing. In: *CVPR* (2017)
16. 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>