Supplementary material:

Joint Bi-layer Optimization for Single-image Rain Streak Removal

There are three parts in this supplementary material. The first part (part 1) gives the optimization details of Subproblem 1.1 in Section 5.4 of the paper. The second part (part 2) presents additional comparison results, while the third part (part 3) presents more results produced from our method.

Part 1. Optimization details of Subproblem 1.1 in Section 5.4 of main paper

In the first part, we present the details on how we use the iterative re-weighted least squares (IRLS) technique to solve Subproblem 1.1 in Section 5.4 of the main paper. The symbols and notations shown below follow those in the main paper.

Subproblem 1.1 on B: By removing $\lambda_1\mathbf{\Psi}(\mathbf{B})$ and adding $\frac{\beta}{2}||\beta^{-1}\mathbf{H}^t||_F^2$ to Eq.10 of the main paper, we can estimate $\mathbf{B}^{t+1}$ as:

$$\min_{\mathbf{B}} ||\mathbf{I}-\mathbf{B}-\mathbf{R}_{k-1}||_F^2 + \lambda_2\Phi(\mathbf{B}) + \frac{\beta}{2}||\mathbf{B}-\mathbf{D}\circ\boldsymbol{\alpha}^t - \frac{1}{\beta}\mathbf{H}^t||_F^2 \; . \tag{1}$$

Now, the optimization is quadratic but has a non-linear term $\mathbf{\Phi}(\mathbf{B})$. Since a quadratic optimization with a quadratic regularization term can be optimized linearly [3, 2, 1], we solve the problem with the iterative re-weighted least squares (IRLS) technique [4]. The main idea here is to decompose the non-linear $\mathbf{\Phi}(\mathbf{B})$ into a quadratic term and another non-linear term (denoted as $\mathbf{S}_{x,i}$), which enable us to solve the optimization in Eq. 1 by computing the new term $\mathbf{S}_{x,i}$ and estimating $\Phi(B)$ with a closed-form solution alternatively. Specifically, we decompose $\mathbf{\Phi}(\mathbf{B})$ for x direction (likewise for y) as

$$\frac{1}{\mathbf{\Theta}(\mathbf{B})_i+\varepsilon_1} \approx \frac{1}{\mathbf{\Theta}(\mathbf{B})_i+\varepsilon_1}\cdot\frac{1}{(\partial_x\mathbf{B})_i^2+\varepsilon_2}\cdot(\partial_x\mathbf{B})_i^2 \; , \tag{2}$$

where $\varepsilon_2 = 0.0001$ is introduced to avoid division by zero. Now, we define scalar term $\mathbf{S}_{x,i}$ (likewise $\mathbf{S}_{y,i}$) as

$$\mathbf{S}_{x,i} = \frac{1}{\mathbf{\Theta}(\mathbf{B})_i+\varepsilon_1}\frac{1}{(\partial_x\mathbf{B})_i^2+\varepsilon_2} \; , \tag{3}$$

such that $\mathbf{\Phi}(\mathbf{B})$ can now be approximated as the average of $\mathbf{S}_{x,i}\,(\partial_x\mathbf{B})_i^2$ and $\mathbf{S}_{y,i}\,(\partial_y\mathbf{B})_i^2$.

Thanks to the approximation, by substituting it into Eq. (1), we can compute $\mathbf{S}_{x,i}$ and $\mathbf{S}_{y,i}$ using current $\mathbf{B}$ to avoid the nonlinear term in Eq. (1). Then, we can take derivative on Eq. (1) and obtain the following sparse linear equation:

$$\left(\mathbf{\Gamma}+\frac{\lambda_2}{2}(\mathbf{E}_x^T V(\mathbf{S}_x)\mathbf{E}_x+\mathbf{E}_y^T V(\mathbf{S}_y)\mathbf{E}_y)\right)V(\mathbf{B}) = V(\mathbf{I}-\mathbf{R}_{k-1}+\mathbf{D}\circ\boldsymbol{\alpha}^t+\frac{1}{\beta}\mathbf{H}^t) \; , \tag{4}$$

where $\mathbf{\Gamma}$ is an identity matrix; $\mathbf{E}_x$ and $\mathbf{E}_y$ are Toeplitz matrices constructed from the discrete gradient values (like [4]); and $V(\cdot)$ is the vectorized operator.

Part 2. Additional Comparison Results

In the second part, we compare our results against results from the following works:

- ***DSC:*** Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, pages 3397–3405, 2015.
- ***GMMLP:*** Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *CVPR*, pages 2736–2744, 2016.
- ***JORDER:*** W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Joint rain detection and rain removal from a single image. In *CVPR*, 2017.
- ***DDN:*** X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *CVPR*, 2017.

Note that for DSC, GMMLP and DDN, we produce the results using the authors' code and tuning its parameters to achieve the best results; while for JORDER, we send input images to the authors and obtain the results directly from them.

(a) *input image*

(b) *DSC*

(c) *GMMLP*

(d) *JORDER*

(e) *DDN*

(f) *our result*

Figure 1: *Additional comparison result #1. (a) input rain photo* (real)*. (b)-(e) results produced from DSC, GMMLP, JORDER and DDN respectively. (f) our result. Obviously, our method has the best performance of removing rain streaks without over-smoothing non-rain details, while DSC, GMMLP, JORDER and DDN retain many rain streaks in their de-rained results.*

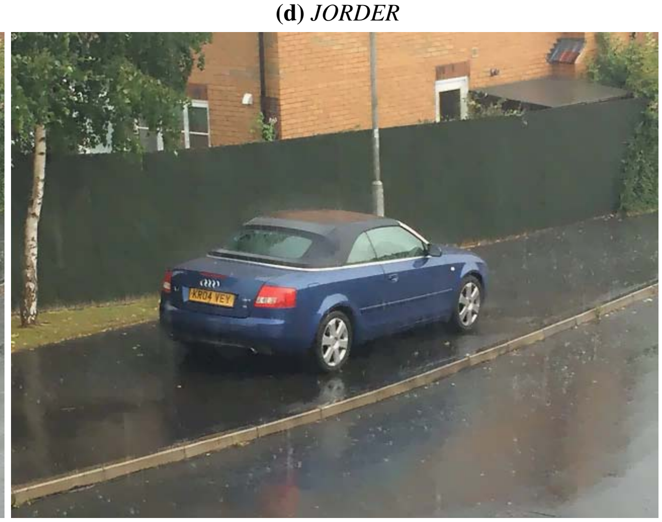
(a) *input image*

(b) *DSC*

(c) *GMMLP*

(d) *JORDER*

(e) *DDN*



(f) *our result*

Figure 2: *Additional comparison result #2. (a) input rain photo (*real*). (b)-(e) results produced from DSC, GMMLP, JORDER and DDN respectively. (f) our result. Obviously, our method has the best performance of removing rain streaks without over-smoothing non-rain details, while DSC, GMMLP, JORDER and DDN retain many rain streaks in their de-rained results.*

(**a**) *input image* (**b**) *DSC*

(**c**) *GMMLP* (**d**) *JORDER*

(**e**) *DDN* (**f**) *our result*

Figure 3: *Additional comparison result #3. (a) input rain photo (*real*). (b)-(e) results produced from DSC, GMMLP, JORDER and DDN respectively. (f) our result. Obviously, our method has the best performance of removing rain streaks without over-smoothing non-rain details, while DSC, GMMLP, JORDER and DDN retain many rain streaks in their de-rained results.*

(a) *input image* **(b)** *DSC*

(c) *GMMLP* **(d)** *JORDER*

(e) *DNN* **(f)** *our result*

Figure 4: *Additional comparison result #4. (a) input rain photo (*real*). (b)-(e) results produced from DSC, GMMLP, JORDER and DDN respectively. (f) our result. Obviously, our method has the best performance of removing rain streaks without over-smoothing non-rain details, while DSC, GMMLP, JORDER and DDN retain many rain streaks in their de-rained results.*

Part 3. Additional De-raining Results

Figure 5: *Additional result #1 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 6: *Additional result #2 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

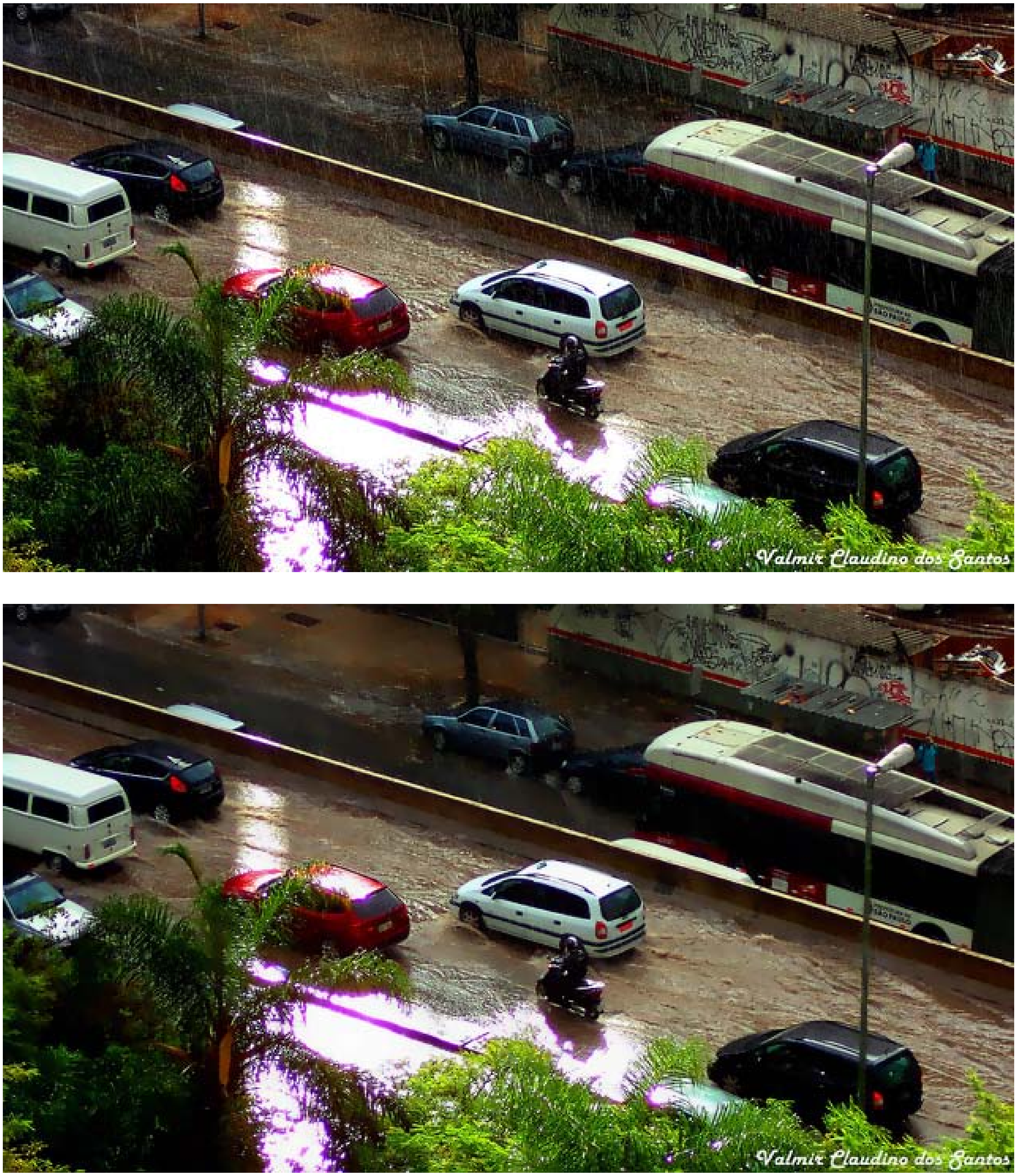


Figure 7: *Additional result #3 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 8: *Additional result #4 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 9: *Additional result #5 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 10: *Additional result #6 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 11: *Additional result #7 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 12: *Additional result #8 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 13: *Additional result #9 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 14: *Additional result #10 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 15: *Additional result #11 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 16: *Additional result #12 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 17: *Additional result #13 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 18: *Additional result #14 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 19: *Additional result #15 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

Figure 20: *Additional result #16 from our method. Top row: input rain photo (real). Bottom row: our result. Our method can effectively remove rain streaks without over-smoothing non-rain details.*

References

[1] D. Krishnan and R. Szeliski. Multigrid and multilevel preconditioners for computational photography. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 30(6):177, 2011. 3

[2] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski. Interactive local adjustment of tonal values. *ACM Transactions on Graphics (SIGGRAPH)*, 25(3):646–653, 2006. 3

[3] R. Szeliski. Locally adapted hierarchical basis preconditioning. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 25, pages 1135–1143. ACM, 2006. 3

[4] L. Xu, Q. Yan, Y. Xia, and J. Jia. Structure extraction from texture via relative total variation. *ACM Trans. Graph. (SIGGRAPH Asia)*, 31(6):139:1–139:10, 2012. 3