WeatherStream: Light Transport Automation of Single Image Deweathering

Howard Zhang[1*] Yunhao Ba[1*] Ethan Yang[1] Varan Mehra[1] Blake Gella[1]
Akira Suzuki[1] Arnold Pfahnl[1] Chethan Chinder Chandrappa[1]
Alex Wong[2] Achuta Kadambi[1]
[1]University of California, Los Angeles [2]Yale University

Supplementary Contents

This supplement is organized as follows:

- Section A illustrates a comparison of synthetic versus real snow weather effects.
- Section B displays additional qualitative results on Internet images.
- Section C shows our pipeline algorithm.
- Section D provides details on the implementation of the pipeline and weather removal models.
- Section E shows intermediate samples from our pipeline.
- Section F shows a failure case of models trained on the WeatherStream Dataset.
- Section G shows some weather removal results of recent GAN models.
- Section H concludes with some closing remarks.

A. Visualization of Synthetic Snow Effects

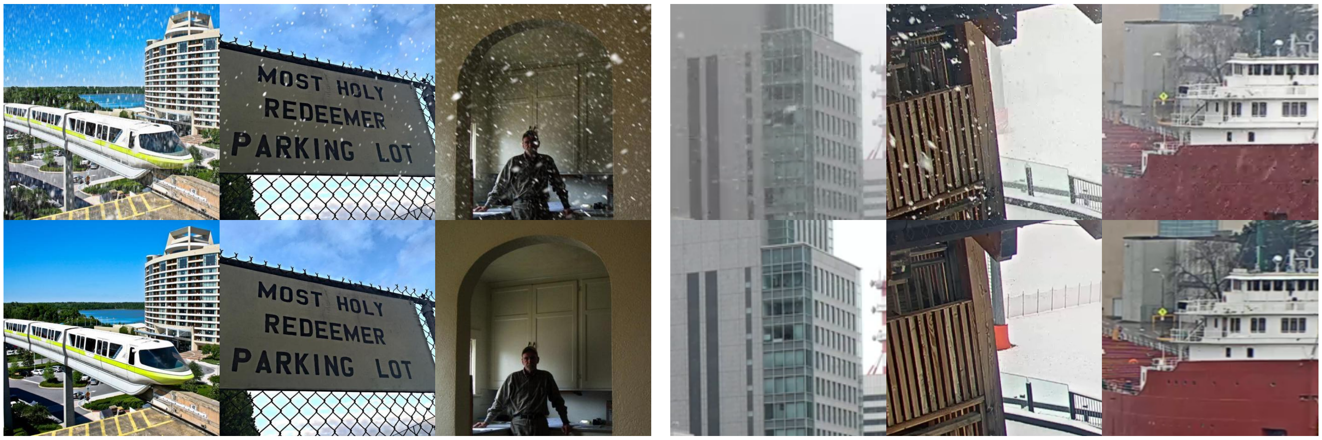
Figure A compares synthetic snow pairs in Snow100K [27] with time-multiplexed pairs in the WeatherStream Dataset. Our dataset captures real snow weather effects with snowflakes and veiling in actual snow environments, while synthetic snow pairs simply add synthetic flakes to images under clear, sunny, or even indoor conditions.

B. Additional Qualitative Results

In addition to Fig. 7, in the main paper, we show some more qualitative results on Internet images of different models trained on a synthetic dataset [39] versus our final WeatherStream Dataset. Figure B compares the results from the Uformer model [41]. Figure C compares the results from the Restormer model [57]. Figure D compares the results from the Transweather model [39]. Figure E compares the results from the Rain-robust model [3].

Models trained on the WeatherStream Dataset are able to preserve more high-frequency details. An example of this can be seen in the second row of Fig. B, where the bike is noticeably sharper on the model trained on WeatherStream Dataset. They are also able to remove more snowflakes, rain streaks, and veiling effects from images, while avoiding blurs such as those found in the building in the third row of Fig. D. Certain more uniquely shaped rain streaks and snowflakes are more effectively removed for models trained on the WeatherStream Dataset. Examples of this are shown in the second row of Fig. C, where all flakes are removed by the model trained on the WeatherStream Dataset. Further examples of this are shown by the second row of Fig. D, where the uniquely motion blurred snowflakes are not removed by the synthetically trained model, but are all removed by the model trained on the WeatherStream Dataset. Models trained on the WeatherStream Dataset are also better able to remove veiling effects from images. An example of this can be seen in the last row of Fig. C and Fig. D, where buildings in the background exhibit higher contrast in the model trained on the WeatherStream Dataset.

*Equal contribution.



Synthetic [27] WeatherStream

Figure A. **WeatherStream Dataset is the first dataset to contain time-multiplexed snow pairs that have real snow effects.** Existing synthetic datasets such as those found in Snow100K [27] cannot model the complexity of real-world snow.

We note that the synthetically trained version of the Rain-robust model [3] is particularly bad at removing rain streaks and snowflakes when compared to training on WeatherStream Dataset. In all rows of Fig. E, it can be seen that a majority of rain streaks and snowflakes remain in the synthetically trained model. This is likely due to the model being designed for training on real weather effect pairs rather than synthetic weather effect pairs.

C. Pipeline Algorithm

We provide in Algorithm 1 the algorithm used for our pipeline, written in pseudo-code, which references equations and sections from the main paper.

Algorithm 1: WeatherStream Collection Pipeline

Input : A candidate set $\mathcal{D}_\mathbf{c}$ of clean frames and a candidate set $\mathcal{D}_\mathbf{d}$ of frames capturing weather effects.
Output: A single good pair of clean and degraded frames $(\mathbf{c}, \mathbf{d})$.
for $(\widetilde{\mathbf{c}}^n, \widetilde{\mathbf{d}}^m)$ *pair in* $\mathcal{D}_\mathbf{c}, \mathcal{D}_\mathbf{d}$ **do**
  Obtain filter mask $\mathbf{B}_{\text{OF}}$ based on Sec. 4, Filtering Block One;
  Obtain filter mask $\mathbf{B}_{\text{static}}$ based on Eq. 14;
  Obtain filter mask $\mathbf{B}_{\text{sky}}$ based on Sec. 4, Filtering Block One;
  Obtain filter mask $\mathbf{B}_{\text{chrom\_var}}$ based on Sec. 4, Filtering Block Two;
  Combine the filter masks $\mathbf{B} \leftarrow \mathbf{B}_{\text{OF}} \cup \mathbf{B}_{\text{static}} \cup \mathbf{B}_{\text{sky}} \cup \mathbf{B}_{\text{chrom\_var}}$;
  Crop the frames based on $\mathbf{B}$: $\widetilde{\mathcal{C}}_c^n \leftarrow \{\widetilde{\mathbf{c}}_{c1}^n, \widetilde{\mathbf{c}}_{c2}^n, ..., \widetilde{\mathbf{c}}_{cn}^n\}$ $\widetilde{\mathcal{D}}_c^m \leftarrow \{\widetilde{\mathbf{d}}_{c1}^m, \widetilde{\mathbf{d}}_{c2}^m, ..., \widetilde{\mathbf{d}}_{cn}^m\}$;
  for $(\widetilde{\mathbf{c}}_c^n, \widetilde{\mathbf{d}}_c^m)$ *pair in* $\widetilde{\mathcal{C}}_c^n, \widetilde{\mathcal{D}}_c^m$ **do**
    grad_check $\leftarrow f_{\text{grad}} > \gamma_{\text{grad}}$ with Eq. 19;
    fft_check $\leftarrow f_{\text{fft}} > \gamma_{\text{fft}}$ with Eq. 20;
    Obtain illum_check based on Sec. 4, Filtering Block Four;
    if grad_check and fft_check and illum_check **then**
      Set $(\mathbf{c}, \mathbf{d}) = (\widetilde{\mathbf{c}}_c^n, \widetilde{\mathbf{d}}_c^m)$;
    end if
  end for
end for
return $(\mathbf{c}, \mathbf{d})$;

Input | Uformer [41] Synthetic | Uformer [41] WeatherStream

Input | Uformer [41] Synthetic | Uformer [41] WeatherStream

Input | Uformer [41] Synthetic | Uformer [41] WeatherStream

Input | Uformer [41] Synthetic | Uformer [41] WeatherStream

Figure B. **Additional qualitative results on Uformer [41].**

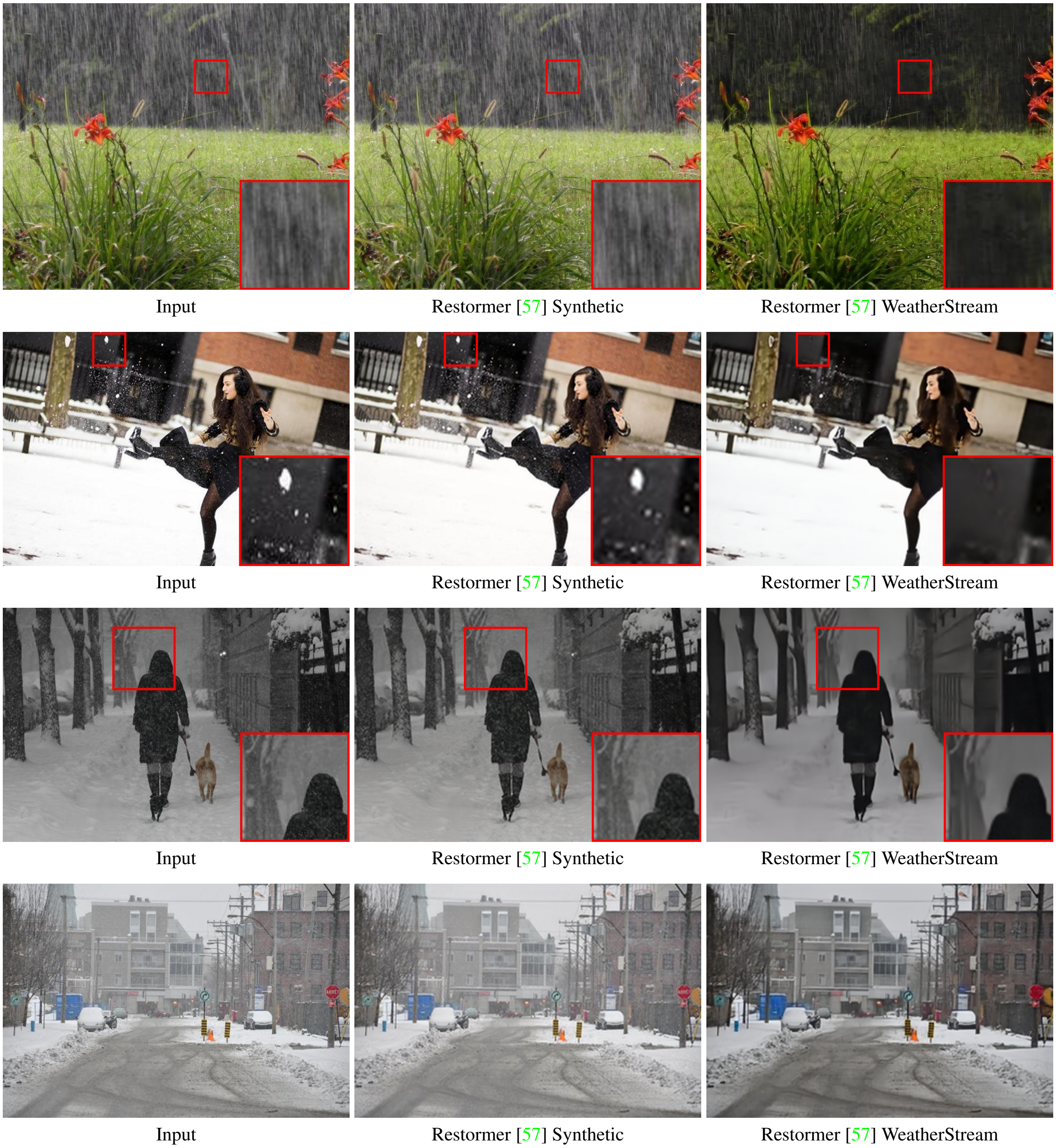


Figure C. **Additional qualitative results on Restormer [57].** Models trained on WeatherStream Dataset are able to remove more snowflakes than those trained on synthetic data.

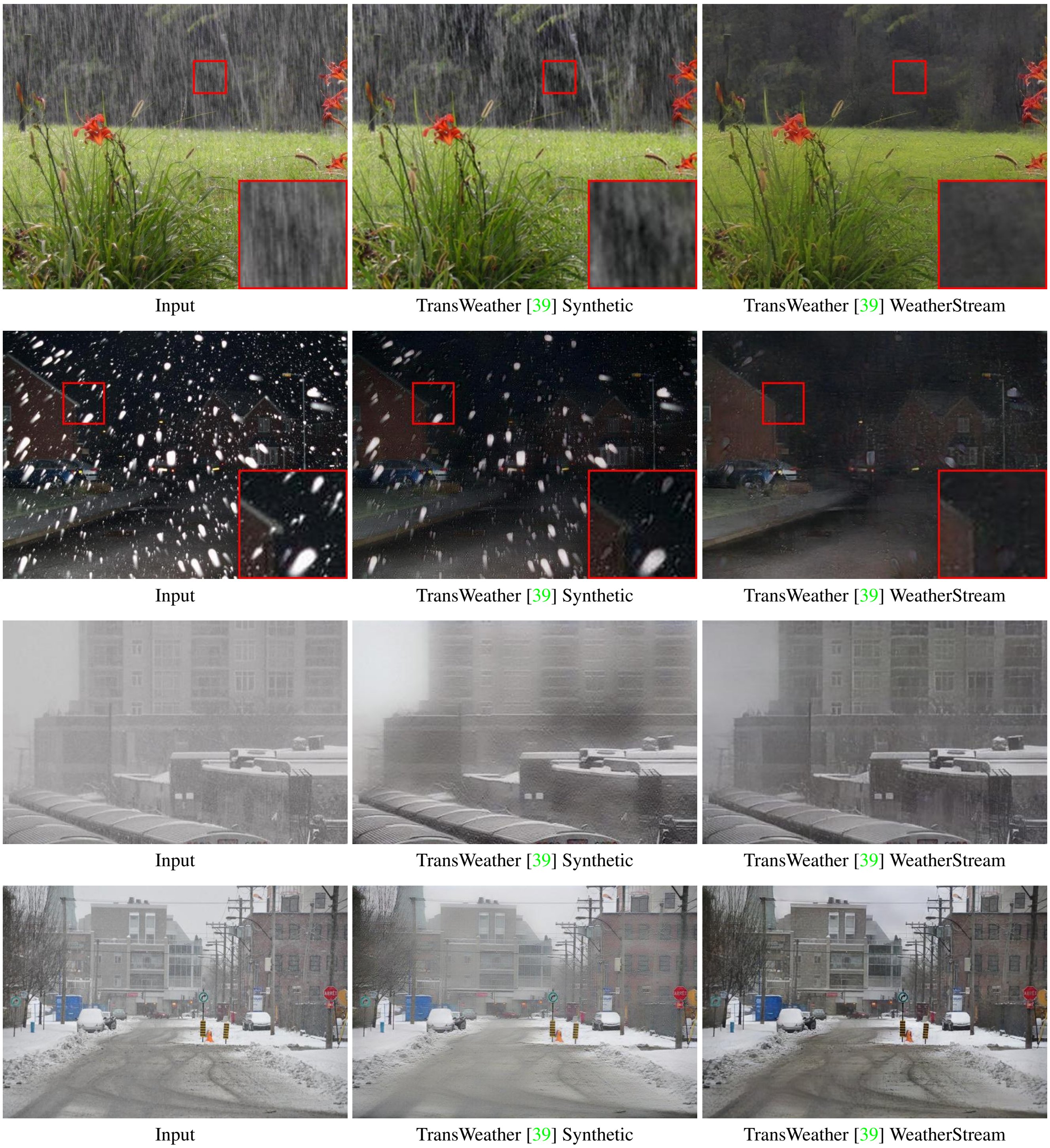
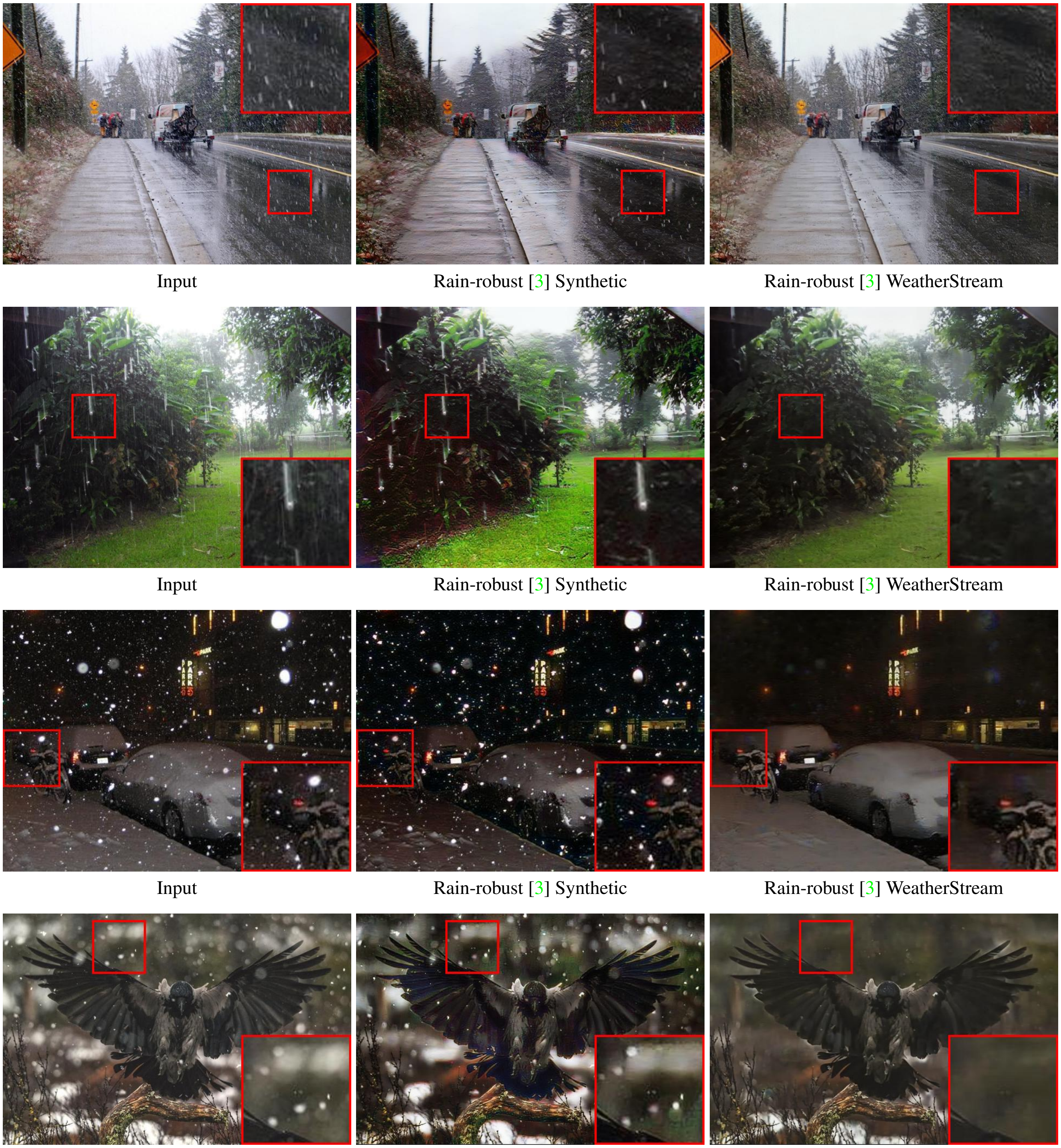


Figure D. **Additional qualitative results on TransWeather [39].** Note the blurry building in the third row result for models trained on synthetic data.



Input | Rain-robust [3] Synthetic | Rain-robust [3] WeatherStream

Input | Rain-robust [3] Synthetic | Rain-robust [3] WeatherStream

Input | Rain-robust [3] Synthetic | Rain-robust [3] WeatherStream

Input | Rain-robust [3] Synthetic | Rain-robust [3] WeatherStream

Figure E. **Qualitative results for the Rain-robust model [3] trained on sythetic data.** The synthetically trained model is unable to remove as many snowflakes and rain streaks.

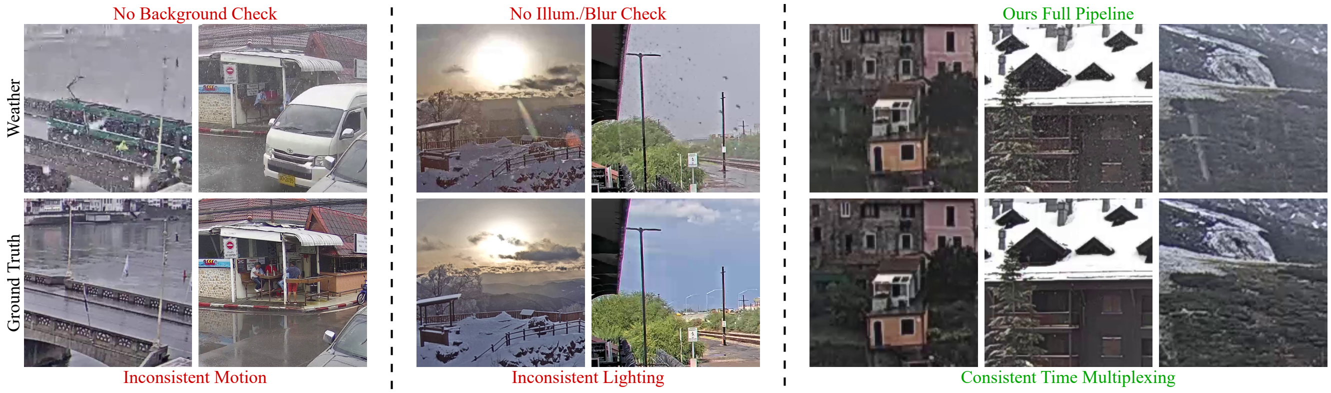


Figure F. **The WeatherStream pipeline filters out inconsistent motion and lighting.** Above images are intermediate outputs from the pipeline after no filtering blocks, the second block, and the final block from left to right.

D. Additional Implementation Details

As an additional supplement to the main paper, we list here more implementation details for our pipeline and the weather removal models included in the main paper.

D.a. Pipeline

The pipeline uses Gunnar-Farneback optical flow [12] with a window size of 25 pixels, and 3 pyramid levels with scale 0.5. Our temporal averaging takes a mean over 15 images. For detecting static object discrepancies without the color verification, we set γ_{static} in Eq. 12 from the main paper to be a PSNR difference of 20dB, which translates to an absolute error of 0.1 for images scaled between 0 and 1. For the FFT-based multi-scatter verification check, we set the center region of the image in the Fourier domain to 0. For the illumination verification check, we use the Rain-robust model [3] as the seed model. This model is trained on a seed dataset including GT-RAIN [3] and a manually collected snow dataset totaling 40K pairs. The snow dataset includes time-multiplexed pairs collected through a similar procedure as GT-RAIN. Hysteresis [6] thresholds for this verification are as follows. All PSNR values are with respect to the time-multiplexed ground truth. If the PSNR of the restored image (using the seed model) is above 25dB, we pass the pair with no further checks. If an illumination shift is present, either the input PSNR is extremely low, or there is no obvious improvement from passing the image through the seed model, since the seed model is able to partially restore the background from veiling effects while leaving the illumination largely unchanged. Therefore, if the PSNR of the restored image is between 20 and 25dB, we only pass the pair if the PSNR increases by at least 2dB between restored and input. If the input PSNR is lower than 20dB, we reject the pair.

D.b. Weather Removal Models

All initial models are trained on the initial seed dataset of 40k pairs as described above. Final models are trained on the entire WeatherStream Dataset. For synthetic models, we use the dataset given by Transweather [39], which includes 9000 synthetic snow and 10069 synthetic rain pairs. The Rain-robust model [3] is trained with a learning rate of 8e-4 and a batch size of 8 on a single NVIDIA 3090 GPU. The Uformer model was trained with learning rate of 2e-4 and a batch size of 32 on 2 NVIDIA 3090 GPUs. The Restormer model was trained with learning rate 3e-4 and a batch size of 6 on a single NVIDIA 3090 GPU. The Transweather model was trained with a learning rate of 2e-4 and a batch size of 32 on 4 NVIDIA 3080 GPUs. Links to the locations of the codebases used for these methods are found in Tab. A.

E. Intermediate Samples from the Pipeline

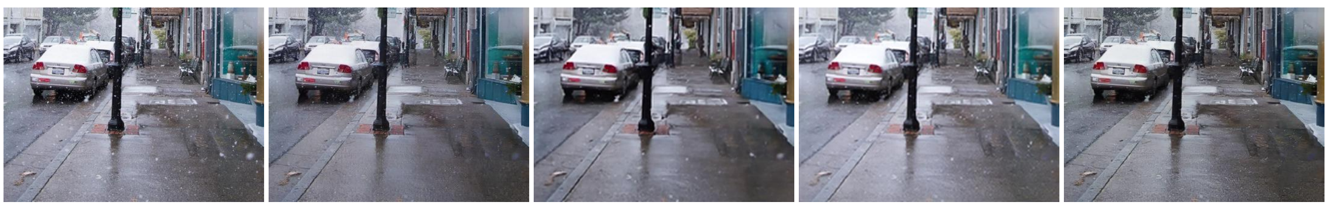
Fig. F shows intermediate outputs from certain stages of the pipeline. For example, with no background consistency check, a train is clearly visible in the weathered image, while it is missing in the ground truth. Without an illumination or blur check, clouds going over the sun can completely change the lighting of the scene between the weathered and ground truth images. Samples taken from the output of our pipeline do not have any of these issues.

Methods	Links
TransWeather [39] (CVPR'22)	https://github.com/jeya-maria-jose/TransWeather
Restormer [57] (CVPR'22)	https://github.com/swz30/Restormer
Uformer [41] (CVPR'22)	https://github.com/ZhendongWang6/Uformer
Rain-robust [3] (ECCV'22)	https://github.com/UCLA-VMG/GT-RAIN

Table A. **Code links for the comparison methods.**

F. Failure Case

We show in Fig. G a case in which some models trained on the WeatherStream Dataset are not able to remove some specific snowflakes. While all models benefit from the larger dataset, certain snowflakes have unusual shapes that are not commonly found in the collected dataset. Therefore, some models may not be able to remove all of these flakes. We expect these results to improve as the dataset is further expanded in future work.



Snowy | TransWeather [39] | Restormer [57] | Uformer [41] | Rain-robust [3]

Figure G. **Failure case.** Performance is dependent on model architecture.

G. GAN Results

Method	Rain	Rain Fog	Snow	Overall
Input	21.38/0.7710	18.36/0.7542	20.70/0.7865	20.18/0.7699
DerainCycleGAN	21.89/0.7733	-	-	-
ZeroScatter	13.17/0.6024	13.70/0.6687	15.41/0.6975	13.99/0.6520
Ours Restormer	**23.67/0.8027**	**22.90/0.8029**	**22.51/0.8279**	**23.08/0.8100**

Table B. **The Restormer model retrained on the WeatherStream dataset outperforms GAN-based methods such as DerainCycle-GAN and ZeroScatter.**

Input | ZeroScatter | Ours/Restormer | Ground Truth

Figure H. **Style translation methods such as ZeroScatter sometimes generate results that exhibit strong color shifts.**

In Tab. B, we compare the Restormer model retrained on the WeatherStream dataset with popular GAN-based weather removal methods such as ZeroScatter [35] and DerainCycleGAN [43]. Only the rain subset is run for DerainCycleGAN, as it is only a deraining method, while ZeroScatter targets all scattering media. For both ZeroScatter and DerainCycleGAN, we use the model weights provided by the authors. The table shows that retraining Restormer on the WeatherStream dataset achieves higher PSNR/SSIM metrics than ZeroScatter and DerainCycleGAN. Note that ZeroScatter exhibits particularly lower results since it sometimes generates outputs with strong color shifts. This is likely due to style translation models' dependence on training data. An example of this is shown in Fig. H.

H. Additional Remarks

WeatherStream is the first large scale time-multiplex dataset for training and evaluating all weather removal (deweathering) models. This is, however, only the first step. While the current revision only supports the deweathering, we further look towards addressing the problem of robust vision under adverse weather by providing additional labeling of the frames to support recognition tasks. Nonetheless, deweathering models trained on WeatherStream are not only suited for appealing to aesthetics via image restoration, but may also support the re-use of pretrained models for downstream vision tasks such as depth completion [15,25,28,29,36,46–49,53,56], stereo [4,7,11,50,55], optical flow [1,20–22,37,38], object detection [16,19,26,33], segmentation [8–10,23,24,34,44,54] and monocular depth prediction [2,5,13,14,17,18,23,30–32,40,42,45,51,52].

References

[1] Filippo Aleotti, Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning end-to-end scene flow by distilling single tasks knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10435–10442, 2020. 9

[2] Yunhao Ba, Alex Gilbert, Franklin Wang, Jinfa Yang, Rui Chen, Yiqin Wang, Lei Yan, Boxin Shi, and Achuta Kadambi. Deep shape from polarization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 554–571. Springer, 2020. 9

[3] Yunhao Ba, Howard Zhang, Ethan Yang, Akira Suzuki, Arnold Pfahnl, Chethan Chinder Chandrappa, Celso de Melo, Suya You, Stefano Soatto, Alex Wong, and Achuta Kadambi. Not just streaks: Towards ground truth for single image deraining. In *European Conference on Computer Vision*. Springer, 2022. 1, 2, 6, 7, 8

[4] Zachary Berger, Parth Agrawal, Tian Yu Liu, Stefano Soatto, and Alex Wong. Stereoscopic universal perturbations across different architectures and datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2022. 9

[5] Ayush Bhandari, Achuta Kadambi, and Ramesh Raskar. *Computational Imaging*. MIT Press, 2022. 9

[6] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 7

[7] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018. 9

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 9

[9] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 9

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 9

[11] Shivam Duggal, Shenlong Wang, Wei-Chiu Ma, Rui Hu, and Raquel Urtasun. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4384–4393, 2019. 9

[12] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003. 7

[13] Xiaohan Fei, Alex Wong, and Stefano Soatto. Geo-supervised visual depth prediction. *IEEE Robotics and Automation Letters*, 4(2):1661–1668, 2019. 9

[14] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. 9

[15] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. *arXiv preprint arXiv:2103.00783*, 2021. 9

[16] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Jiacong Fang, imyhxy, Kalen Michael, Lorna, Abhiram V, Diego Montes, Jebastin Nadar, Laughing, tkianai, yxNONG, Piotr Skalski, Zhiqiang Wang, Adam Hogan, Cristi Fati, Lorenzo Mammana, AlexWang1900, Deep Patel, Ding Yiwei, Felix You, Jan Hajek, Laurentiu Diaconu, and Mai Thanh Minh. ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, Feb. 2022. 9

[17] Achuta Kadambi, Vage Taamazyan, Boxin Shi, and Ramesh Raskar. Polarized 3d: High-quality depth sensing with polarization cues. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3370–3378, 2015. 9

[18] Achuta Kadambi, Vage Taamazyan, Boxin Shi, and Ramesh Raskar. Depth sensing using geometrically constrained polarization normals. *International Journal of Computer Vision*, 125:34–51, 2017. 9

[19] Agastya Kalra, Guy Stoppi, Bradley Brown, Rishav Agarwal, and Achuta Kadambi. Towards rotation invariance in object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3530–3540, 2021. 9

[20] Dong Lao and Ganesh Sundaramoorthi. Minimum delay moving object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4250–4259, 2017. 9

[21] Dong Lao and Ganesh Sundaramoorthi. Extending layered models to 3d motion. In *Proceedings of the European conference on computer vision (ECCV)*, pages 435–451, 2018. 9

[22] Dong Lao and Ganesh Sundaramoorthi. Minimum delay object detection from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5097–5106, 2019. 9

[23] Dong Lao, Alex Wong, and Stefano Soatto. Does monocular depth estimation provide better pre-training than classification for semantic segmentation? *arXiv preprint arXiv:2203.13987*, 2022. 9

[24] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 82–92, 2019. 9

[25] Tian Yu Liu, Parth Agrawal, Allison Chen, Byung-Woo Hong, and Alex Wong. Monitored distillation for positive congruent depth completion. *arXiv preprint arXiv:2203.16034*, 2022. 9

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 9

[27] Yun-Fu Liu, Da-Wei Jaw, Shih-Chia Huang, and Jenq-Neng Hwang. Desnownet: Context-aware deep network for snow removal. *IEEE Transactions on Image Processing*, 27(6):3064–3073, 2018. 1, 2

[28] Nathaniel Merrill, Patrick Geneva, and Guoquan Huang. Robust monocular visual-inertial depth completion for embedded systems. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2021. 9

[29] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In-So Kweon. Non-local spatial propagation network for depth completion. In *European Conference on Computer Vision, ECCV 2020*. European Conference on Computer Vision, 2020. 9

[30] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3227–3237, 2020. 9

[31] Matteo Poggi, Fabio Tosi, Filippo Aleotti, and Stefano Mattoccia. Real-time self-supervised monocular depth estimation without gpu. *IEEE Transactions on Intelligent Transportation Systems*, 2022. 9

[32] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 9

[33] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 9

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 9

[35] Zheng Shi, Ethan Tseng, Mario Bijelic, Werner Ritter, and Felix Heide. Zeroscatter: Domain transfer for long distance imaging and vision through scattering media. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3476–3486, 2021. 8

[36] Akash Deep Singh, Yunhao Ba, Ankur Sarker, Howard Zhang, Achuta Kadambi, Stefano Soatto, Mani Srivastava, and Alex Wong. Depth estimation from camera image and mmwave radar point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 9

[37] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 9

[38] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 9

[39] Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M Patel. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2353–2363, 2022. 1, 5, 7, 8

[40] Alexander Vilesov, Pradyumna Chari, Adnan Armouti, Anirudh Bindiganavale Harish, Kimaya Kulkarni, Ananya Deoghare, Laleh Jalilian, and Achuta Kadambi. Blending camera and 77 ghz radar sensing for equitable, robust plethysmography. *ACM Trans. Graph.(SIGGRAPH)*, 41(4):1–14, 2022. 9

[41] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17683–17693, 2022. 1, 3, 8

[42] Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2162–2171, 2019. 9

[43] Yanyan Wei, Zhao Zhang, Yang Wang, Mingliang Xu, Yi Yang, Shuicheng Yan, and Meng Wang. Deraincyclegan: Rain attentive cyclegan for single image deraining and rainmaking. *IEEE Transactions on Image Processing*, 30:4788–4801, 2021. 8

[44] Alex Wong, Allison Chen, Yangchao Wu, Safa Cicek, Alexandre Tiard, Byung-Woo Hong, and Stefano Soatto. Small lesion segmentation in brain mris with subpixel embedding. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 7th International Workshop, BrainLes 2021, Held in Conjunction with MICCAI 2021, Virtual Event, September 27, 2021, Revised Selected Papers, Part I*, pages 75–87. Springer, 2022. 9

[45] Alex Wong, Safa Cicek, and Stefano Soatto. Targeted adversarial perturbations for monocular depth prediction. *Advances in Neural Information Processing Systems*, 33, 2020. 9

[46] Alex Wong, Safa Cicek, and Stefano Soatto. Learning topology from synthetic data for unsupervised depth completion. *IEEE Robotics and Automation Letters*, 6(2):1495–1502, 2021. 9

[47] Alex Wong, Xiaohan Fei, Byung-Woo Hong, and Stefano Soatto. An adaptive framework for learning unsupervised depth completion. *IEEE Robotics and Automation Letters*, 6(2):3120–3127, 2021. 9

[48] Alex Wong, Xiaohan Fei, and Stefano Soatto. Voiced: Depth completion from inertial odometry and vision. *ArXiv, abs/1905.08616*, 2019. 9

[49] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters*, 2020. 9

[50] Alex Wong, Mukund Mundhra, and Stefano Soatto. Stereopagnosia: Fooling stereo networks with adversarial perturbations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2879–2888, 2021. 9

[51] Alex Wong and Stefano Soatto. Bilateral cyclic constraint and adaptive regularization for learning a monocular depth prior. 2018. 9

[52] Alex Wong and Stefano Soatto. Bilateral cyclic constraint and adaptive regularization for unsupervised monocular depth prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5644–5653, 2019. 9

[53] Alex Wong and Stefano Soatto. Unsupervised depth completion with calibrated backprojection layers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12747–12756, 2021. 9

[54] Alex Wong and Alan L Yuille. One shot learning via compositions of meaningful patches. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1197–1205, 2015. 9

[55] Haofei Xu and Juyong Zhang. Aanet: Adaptive aggregation network for efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1959–1968, 2020. 9

[56] Yanchao Yang, Alex Wong, and Stefano Soatto. Dense depth posterior (ddp) from single image and sparse range. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3353–3362, 2019. 9

[57] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5728–5739, 2022. 1, 4, 8