**WeatherStream: Light Transport Automation of Single Image Deweathering**

Howard Zhang¹* Yunhao Ba¹* Ethan Yang¹ Varan Mehra¹ Blake Gella¹
Akira Suzuki¹ Arnold Pfahnl¹ Chethan Chinder Chandrappa¹
Alex Wong² Achuta Kadambi¹
¹University of California, Los Angeles ²Yale 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.
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 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**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th>for (c&lt;sup&gt;n&lt;/sup&gt;, d&lt;sup&gt;m&lt;/sup&gt;) pair in D&lt;sub&gt;c&lt;/sub&gt;, D&lt;sub&gt;d&lt;/sub&gt; do</th>
</tr>
</thead>
<tbody>
<tr>
<td>A candidate set D&lt;sub&gt;c&lt;/sub&gt; of clean frames and a candidate set D&lt;sub&gt;d&lt;/sub&gt; of frames capturing weather effects.</td>
<td>A single good pair of clean and degraded frames (c, d),</td>
<td></td>
</tr>
<tr>
<td>for (e&lt;sub&gt;cn&lt;/sub&gt;, e&lt;sub&gt;dm&lt;/sub&gt;) pair in D&lt;sub&gt;c&lt;/sub&gt;, D&lt;sub&gt;d&lt;/sub&gt; do</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Obtain filter mask B&lt;sub&gt;OF&lt;/sub&gt; based on Sec. 4, Filtering Block One;</td>
<td>Obtain filter mask B&lt;sub&gt;static&lt;/sub&gt; based on Eq. 14;</td>
<td></td>
</tr>
<tr>
<td>Obtain filter mask B&lt;sub&gt;static&lt;/sub&gt; based on Eq. 14;</td>
<td>Obtain filter mask B&lt;sub&gt;sky&lt;/sub&gt; based on Sec. 4, Filtering Block One;</td>
<td></td>
</tr>
<tr>
<td>Obtain filter mask B&lt;sub&gt;chrom&lt;/sub&gt; based on Sec. 4, Filtering Block Two;</td>
<td>Combine the filter masks B ← B&lt;sub&gt;OF&lt;/sub&gt; ∪ B&lt;sub&gt;static&lt;/sub&gt; ∪ B&lt;sub&gt;sky&lt;/sub&gt; ∪ B&lt;sub&gt;chrom&lt;/sub&gt;;</td>
<td></td>
</tr>
<tr>
<td>Crop the frames based on B: c&lt;sub&gt;cn&lt;/sub&gt; ← {e&lt;sub&gt;cn&lt;/sub&gt;1, e&lt;sub&gt;cn&lt;/sub&gt;2, ..., e&lt;sub&gt;cn&lt;/sub&gt;n} D&lt;sub&gt;c&lt;/sub&gt; ← {d&lt;sub&gt;cm&lt;/sub&gt;1, d&lt;sub&gt;cm&lt;/sub&gt;2, ..., d&lt;sub&gt;cm&lt;/sub&gt;n};</td>
<td>for (e&lt;sub&gt;cn&lt;/sub&gt;, e&lt;sub&gt;dm&lt;/sub&gt;) pair in c&lt;sub&gt;cn&lt;/sub&gt;, D&lt;sub&gt;c&lt;/sub&gt; do</td>
<td></td>
</tr>
<tr>
<td>grad_check ← f&lt;sub&gt;grad&lt;/sub&gt; &gt; γ&lt;sub&gt;grad&lt;/sub&gt; with Eq. 19;</td>
<td>Obtain illum_check based on Sec. 4, Filtering Block Four;</td>
<td></td>
</tr>
<tr>
<td>fft_check ← f&lt;sub&gt;fft&lt;/sub&gt; &gt; γ&lt;sub&gt;fft&lt;/sub&gt; with Eq. 20;</td>
<td>if grad_check and fft_check and illum_check then</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set (c, d) = (e&lt;sub&gt;cn&lt;/sub&gt;, d&lt;sub&gt;cm&lt;/sub&gt;);</td>
<td></td>
<td></td>
</tr>
<tr>
<td>end if</td>
<td></td>
<td></td>
</tr>
<tr>
<td>end for</td>
<td></td>
<td></td>
</tr>
<tr>
<td>end for</td>
<td></td>
<td></td>
</tr>
<tr>
<td>return (c, d);</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.
Figure E. **Qualitative results for the Rain-robust model [3] trained on synthetic data.** The synthetically trained model is unable to remove as many snowflakes and rain streaks.
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 \( \gamma_{\text{static}} \) 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.
<table>
<thead>
<tr>
<th>Methods</th>
<th>Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restormer [57] (CVPR’22)</td>
<td><a href="https://github.com/swz30/Restormer">https://github.com/swz30/Restormer</a></td>
</tr>
</tbody>
</table>

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.

![Figure G](image)

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

G. GAN Results

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

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

![Figure H](image)

**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, 44, 54] and monocular depth prediction [2, 5, 13, 14, 17, 18, 23, 30–32, 40, 42, 45, 51, 52].

References

Zheng Shi, Ethan Tseng, Mario Bijelic, Werner Ritter, and Felix Heide. Zeroscatter: Domain transfer for long distance imaging and 


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/CVF International Conference on Computer Vision*, 2019. 9


