Robust Unfolding Network for HDR Imaging with Modulo Cameras (Supplementary Material)

1. Proof of Proposition 1

Let $\mathcal{M} : \mathbb{R} \to [0, a)$ denote the modulo operator defined by

$$\mathcal{M}(y) = y \bmod a.$$

Consider $\boldsymbol{X} = \mathcal{M}(\boldsymbol{Y})$, we have then

$$\boldsymbol{X}_{i,j} = \boldsymbol{Y}_{i,j} \bmod a$$

or equivalently,

$$\boldsymbol{X}_{i,j} = \boldsymbol{Y}_{i,j} + a \cdot \boldsymbol{R}_{i,j}, \quad \boldsymbol{R}_{i,j} \in \mathbb{Z}. \tag{1}$$

Applying the partial difference along x-axis on both sides of the Eq. (1), we have

$$\nabla_x \boldsymbol{X}_{i,j} = \nabla_x \boldsymbol{Y}_{i,j} + a \cdot \nabla_x \boldsymbol{R}_{i,j}, \tag{2}$$

where $\boldsymbol{R} \in \mathbb{Z}^{H \times W}$ is an integer matrix.

Next, for any point satisfying $\|\boldsymbol{\nabla} \boldsymbol{Y}_{i,j}\|_\infty < a/2$, we have

$$|\nabla_x \boldsymbol{Y}_{i,j}| < \frac{a}{2},$$

which gives

$$0 < \nabla_x \boldsymbol{Y}_{i,j} + \frac{a}{2} < a. \tag{3}$$

Define $\widetilde{\mathcal{M}}$ as

$$\widetilde{\mathcal{M}}(y) = \mathcal{M}(y + \frac{a}{2}) - \frac{a}{2} = \big[(y + \frac{a}{2}) \bmod a\big] - \frac{a}{2}.$$

Notice that $\nabla_x \boldsymbol{R}_{i,j}$ is an integer since $\boldsymbol{R}$ is an integer matrix. We have then

$$\begin{array}{rcl} \widetilde{\mathcal{M}}\big(\nabla_x \boldsymbol{X}_{i,j}\big) & = & \mathcal{M}\big(\nabla_x \boldsymbol{Y}_{i,j} + \frac{a}{2} + a \cdot \nabla_x \boldsymbol{R}_{i,j}\big) - \frac{a}{2} \\ & = & \mathcal{M}\big(\nabla_x \boldsymbol{Y}_{i,j} + \frac{a}{2}\big) - \frac{a}{2} \\ & = & \big(\nabla_x \boldsymbol{Y}_{i,j} + \frac{a}{2}\big) - \frac{a}{2} \qquad \big[\text{by Eq. (3)}\big] \\ & = & \nabla_x \boldsymbol{Y}_{i,j}. \end{array}$$

The same argument is also applicable to $\nabla_y \boldsymbol{X}_{i,j}$. Therefore, we have that

$$\widetilde{\mathcal{M}}(\boldsymbol{\nabla} \boldsymbol{X}_{i,j}) = \boldsymbol{\nabla} \boldsymbol{Y}_{i,j}, \ \text{ if } \ \|\boldsymbol{\nabla} \boldsymbol{Y}_{i,j}\|_\infty = \max\{|\nabla_x \boldsymbol{Y}_{i,j}|, |\nabla_y \boldsymbol{Y}_{i,j}|\} < a/2. \tag{4}$$

The proof is done.

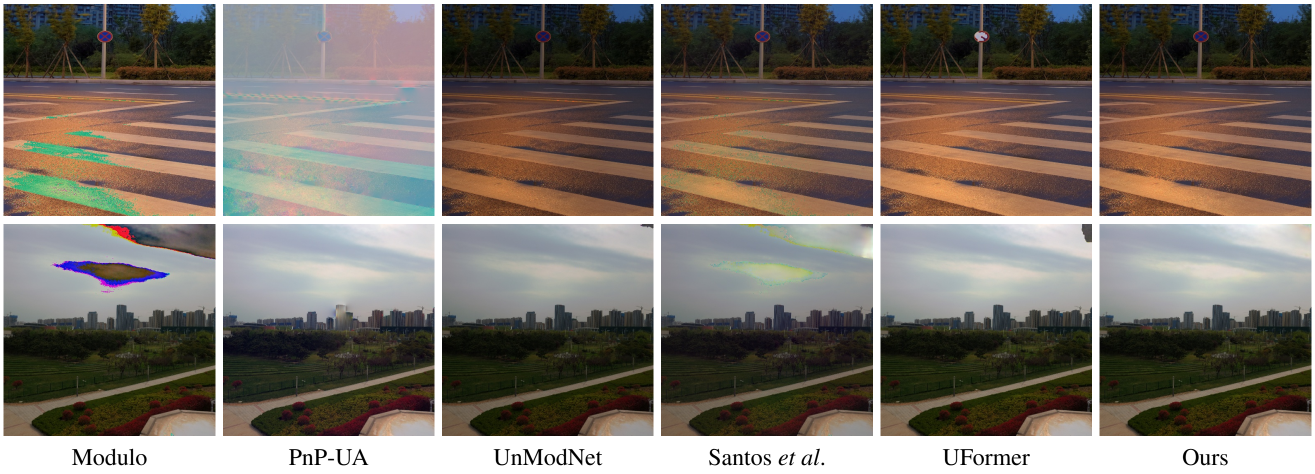
Modulo PnP-UA UnModNet Santos *et al.* UFormer Ours

Figure 1. Qualitative results of the compared methods on two modulo samples of real-RGB dataset.

2. More Visual Results on Real-RGB Dataset

This section provides additional qualitative comparison of our DUN and other methods on the real-RGB dataset, as shown in Fig. 1. From the figure, PnP-UA fails to recover accurate illumination or color, while the method Santos *et al*. produces modulo patterns on the road crossing (upper case) and in the sky (bottom case). UFormer shows improvement but still suffers from some unnatural distortions. UnModNet visually surpasses the aforementioned methods but introduces overall darkness or moderate artifact. In contrast, our DUN provides more authentic visual results for both samples.

3. Visual Evaluation on Real-sensor Dataset

We evaluate our DUN, PnP-UA and UnModNet which are specifically designed for modulo HDR reconstruction, on the sample of real-sensor dataset [2]. This dataset collects 8-bit grayscale modulo images. See the visual comparison in Fig. 2. Similar to the results in real-RGB dataset, PnP-UA exhibits significant illumination discrepancy in its reconstruction. While both UnModNet and our method produce visually plausible results, our DUN demonstrates better visual quality with fewer artifact (*e.g*., modulo pattern on the spherical structure in the result of UnModNet).

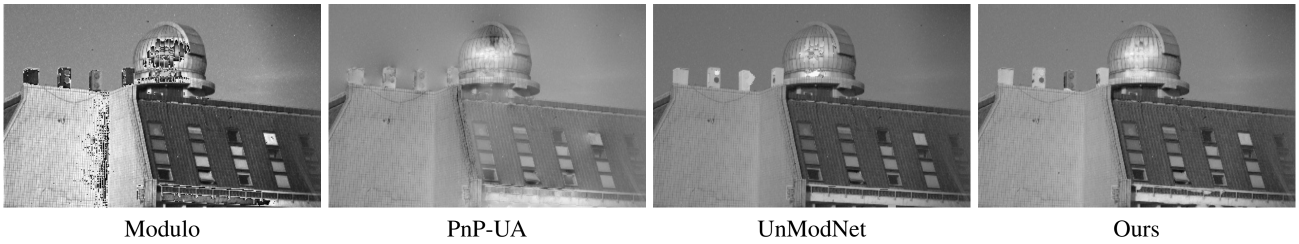
Modulo PnP-UA UnModNet Ours

Figure 2. Qualitative results of the compared methods on the modulo sample of real-sensor dataset.

4. Evaluation on Noisy Modulo Image Datasets

We evaluate our DUN and other compared methods under noisy scenarios. Following the same pipeline described in the manuscript, we generate 4,000 HDR ground truth images from 400 randomly-selected images in the dataset. Their noisy modulo versions are synthesized via [1]: $\boldsymbol{X} = \mathcal{M}(\boldsymbol{Y} + \boldsymbol{N})$, where $\boldsymbol{N} \in \mathbb{R}^{H\times W\times C}$ denotes the noise. Specifically, we introduce additive white Gaussian noise (AWGN) with SNR levels of 20 dB and 30 dB to create two datasets for training, respectively. For test, the remaining images are processed similarly to generate HDR images and their corrupted modulo counterparts with the same SNR levels.

The quantitative performance results of the compared methods are listed in Tab. 1. Our DUN ranks the top across all metrics in both noisy settings. The visual comparison, shown in Fig. 3, further confirms the effectiveness of our method, where

Table 1. Quantitative performance results of the compared methods for noisy modulo HDR. The **Best** results are marked in **Bold**.

Method	SNR = 20 dB				SNR = 30 dB			
	NRMSE(%)	PSNR(dB)	SSIM	MS-SSIM	NRMSE(%)	PSNR(dB)	SSIM	MS-SSIM
MRF	45.46	24.76	0.38	0.54	45.42	24.78	0.38	0.54
PnP-UA	31.63	29.04	0.34	0.50	28.76	29.25	0.34	0.50
UnModNet	12.97	30.81	0.63	0.80	10.54	32.96	0.79	0.88
ExpandNet	12.80	22.50	0.71	0.71	10.28	23.44	0.79	0.77
Santos *et al.*	9.75	36.26	0.90	0.91	9.52	37.35	0.94	0.95
UFormer	8.43	40.44	0.98	0.97	7.53	42.32	0.98	0.98
Ours	**6.71**	**40.52**	**0.98**	**0.97**	**5.96**	**42.45**	**0.98**	**0.98**

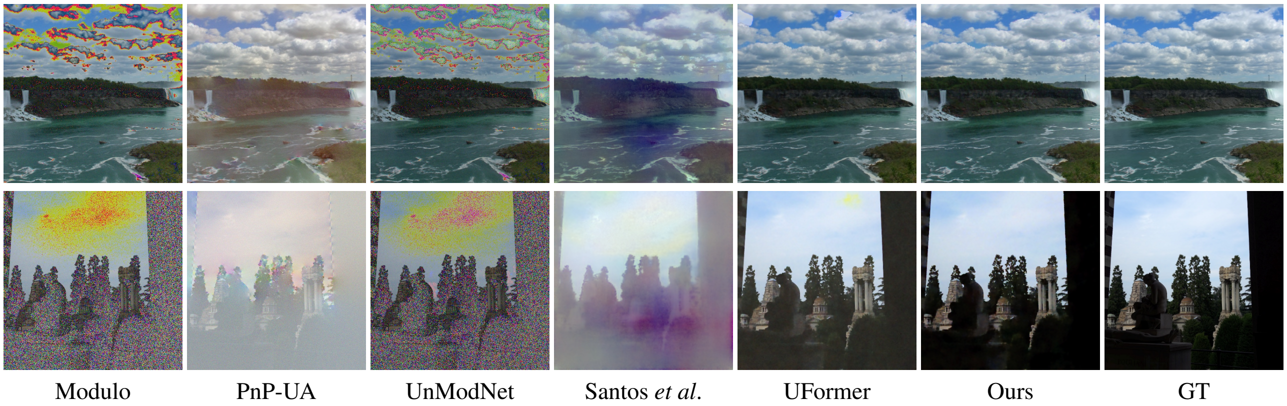

Figure 3. Visual results of compared methods on two modulo samples with different noise levels (Top: SNR=30dB; Bottom: SNR=20dB).

our DUN consistently outperforms other methods in both samples and provides noise-resistant reconstructions with details. It is unsurprising since our superiority stems from the adaptive subtraction mechanism by an auxiliary variable $\boldsymbol{V}$, which simultaneously compensates for outlier-induced residuals and eliminates the noises from corrupted gradients. Although the performance gaps between our method and UFormer are narrower compared to that of noise-free scenario, our DUN maintains much smaller parameter count and FLOPs compared to UFormer, demonstrating that our DUN delivers competitive performance with significantly higher computational efficiency. In contrast, UnModNet suffers notable performance drops compared to that of clean dataset (*e.g.*, 32.96 dB *v.s.* 40.42 dB). This is mainly because its framework of predicting the rollover counts is incapable of removing the noise.

5. More Implementation Details of the DUN

Our DUN adopts a three-phase ($K = 3$) unfolding architecture with the step number in AGD set as $N = 10$.

Details about the UNet$_{\boldsymbol{Y}}$**:** The first convolutional layer in the UNet$_{\boldsymbol{Y}}$ generates 16-channel feature maps, followed by progressive channel expansion to 32, 64, 128, and 256 channels as the spatial resolution halves through strided convolution. A decoder with bilinear upsampling layers generates the features with consistent channel numbers as those of the downscaling encoder.

Details about the spiking neuron-based module: In the spiking neuron-based module, we employs four spiking neurons ($L = 4$) for the prediction of binary map for $\boldsymbol{V}_k$. The synaptic weight applied to the output of previous neuron (*i.e.*, $\boldsymbol{W}_l \cdot \boldsymbol{O}_{l-1}^{(t)}$) is implemented via a convolutional layer followed by PReLU activation. The hidden channel numbers and kernel sizes of all convolutional layers in this module are set to 16 and 3×3, respectively.

Configuration of parameters: The learnable step size η_k for gradient descent and thresholding value τ are initially set to 0.1 and 0.5, respectively. The learnable weight w_k for summation in UNet$_{\boldsymbol{Y}}$ is initialized as $0.1 \times 3^{k-1}$ for the k^{th} phase of DUN. The parameters κ and p remain constant values of 0.5 and 1 throughout training, respectively.

6. Inference Flow of the DUN

Let SNM denote the spiking neuron-based module, the inference flow of our proposed DUN is formulated in algorithm 1.

Algorithm 1 Inference flow of the DUN.

```
Require: $\boldsymbol{X}$: modulo image
Ensure: $\boldsymbol{Y}_K$: recovered HDR image
$\boldsymbol{Y}_0 = 2^B \boldsymbol{I}$, $\boldsymbol{V}_0 = \boldsymbol{0}$
for $k = 1$ to $K$ do
    $\boldsymbol{Q}_k^{(0)} = \boldsymbol{P}_k^{(0)} = \boldsymbol{Y}_{k-1}, \beta^{(0)} = 1$
    for $n = 1$ to $N$ do
        $\boldsymbol{Q}_k^{(n)} = \boldsymbol{P}_k^{(n-1)} + \eta_k \mathcal{G}_{\boldsymbol{P}} \big( \boldsymbol{\nabla} \boldsymbol{P}_k^{(n-1)} - (\widetilde{\mathcal{M}}(\boldsymbol{\nabla} \boldsymbol{X}) - \boldsymbol{V}_{k-1}) \big)$
        $\beta^{(n)} = \big( 1 + \sqrt{1 + 4\beta^{(n-1)} \cdot \beta^{(n-1)}} \big) / 2$
        $\boldsymbol{P}_k^{(n)} = \boldsymbol{Q}_k^{(n)} + \frac{\beta^{(n-1)} - 1}{\beta^{(n)}} (\boldsymbol{Q}_k^{(n)} - \boldsymbol{Q}_k^{(n-1)})$
    end for
    $\boldsymbol{Y}_k = \mathrm{UNet}_{\boldsymbol{Y}} (\boldsymbol{Q}_k^{(N)}, \boldsymbol{X})$
    $\boldsymbol{E}_k = \widetilde{\mathcal{M}}(\boldsymbol{\nabla} \boldsymbol{X}) - \boldsymbol{\nabla} \boldsymbol{Y}_k$
    $\boldsymbol{V}_k = \mathrm{SNM}(\boldsymbol{E}_1, \cdots, \boldsymbol{E}_k)$
end for
```

References

[1] Jorge Bacca, Brayan Monroy, and Henry Arguello. Deep plug-and-play algorithm for unsaturated imaging. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2460–2464. IEEE, 2024. 2

[2] Chu Zhou, Hang Zhao, Jin Han, Chang Xu, Chao Xu, Tiejun Huang, and Boxin Shi. Unmodnet: Learning to unwrap a modulo image for high dynamic range imaging. *Advances in Neural Information Processing Systems*, 33:1559–1570, 2020. 2