RIFLE: Removal of Image Flicker-Banding via Latent Diffusion Enhancement

Libo Zhu[1*], **Zihan Zhou**[1*], **Xiaoyang Liu**[1], **Zhiyi Zhou**[1],
Weihang Zhang[2], **Keyu Shi**[2], **Yifan Fu**[2], **Yulun Zhang**[1†]
[1]Shanghai Jiao Tong University, [2]Institute of Media Technology and Experience Design, Huawei

Abstract

Capturing screens is common, but photos of emissive displays are often influenced by *flicker-banding* (FB), some alternating bright–dark stripes due to temporal aliasing between a camera's rolling-shutter readout and display's brightness modulation. Unlike moiré degradation, FB remains underexplored despite its frequent and severe impact on readability and perceived quality. We formulate FB removal as a dedicated restoration task and introduce **R**emoval of **I**mage **F**licker-**B**anding via **L**atent Diffusion **E**nhancement, ***RIFLE***, a diffusion-based framework designed to remove FB with fine details. We introduce a task-specific ***Banding-Suppressed High-Frequency Prior (BSHP)***, which combines gradient-based high-frequency localization with a smooth structural support map to build a compact restoration prior, and injects it into the restoration backbone via multi-stage FiLM modulation. Moreover, a task-specific ***Masked Loss (ML)*** uses the mask from FB simulation to concentrate supervision on banded regions while preserving global fidelity. To overcome data scarcity, we provide a ***simulation pipeline*** synthesizing FB in the luminance domain with stochastic jitter in banding angle, spacing, and width. Feathered boundaries and sensor noise are also applied for a more realistic simulation. For evaluation, we collect a ***paired real-world FB dataset*** with pixel-aligned banding-free references captured via long exposure. Across quantitative metrics and visual comparisons on our real-world dataset, RIFLE consistently outperforms recent image reconstruction baselines. To our knowledge, this is ***the first work*** on FB simulation and removal. Our dataset and code will be released at `https://github.com/libozhu03/RIFLE`

1 Introduction

Capturing screens is routine in our everyday life: (i) students photograph lecture slides on projectors in class, (ii) professionals document dashboards and error messages on monitors, (iii) creators share phone or smartwatch interfaces, (iv) commuters record LED billboards or vehicle clusters, etc. Despite impressive progress in mobile imaging, photos of emissive displays remain prone to characteristic degradations. Among these, moiré [1–5] from spatial interference between the subpixel display lattice and the camera sampling grid has been widely studied, producing effective learning-based remedies. However, our empirical survey of real-world captures reveals a different, underexplored image degradation that frequently dominates visual quality: ***Flicker-banding***.

Flicker-banding (FB) appears as alternating bright and dark stripes that traverse images, often approximately horizontal but sometimes tilted or warped. The cause is temporal: most commodity smartphone sensors employ electronic rolling shutters that expose rows sequentially, while modern displays modulate luminance in time via pulse-width modulation (PWM) or scanning refresh.

*Equal contribution
†Corresponding authors: Yulun Zhang, yulun100@gmail.com

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



Figure 1: Overview of ***RIFLE***. The top part shows the datasets we build, including the simulated training dataset (first row) and the real-world testing dataset (second row). The bottom part shows the flicker-banding removal effect on the simulated datasets and real-world scenarios.

When the camera's line readout cadence aliases the display's time-varying emission, the temporal mismatch collapses into spatial striping within a single frame. The perceptual impact is so substantial that FB largely distracts attention, obscures screen elements and small fonts, and distorts tone and contrast. FB often degrades the perceived quality of screen photographs and may result in an unsatisfactory viewing experience, especially when small screen elements must remain legible.

Removing FB is an inherent challenge for three reasons. (i) ***Missing screen-side metadata.*** During photography, the camera has no access to the screen's driving parameters (*e.g.*, PWM frequency, duty cycle, scanning order), which are device dependent and mode dependent. Therefore, hardware-side FB mitigation solutions are extremely hard. (ii) ***Various morphological types.*** Stripe orientation, spacing, duty cycle, and contrast depend jointly on the sensor readout speed, exposure, gain, and display technology and settings. Different parameters yield diverse and non-stationary patterns that strain single-prior restorers. (iii) ***Partial information loss.*** In severe cases, the dark phase of the modulation produces near-black bars that overwrite scene content. Successful restoration must reason about missing structures, not merely denoise or deblur. All these factors differentiate FB from the classic moiré or global deflicker and call for a task-specific image reconstruction solution.

These challenges motivate a dedicated learning-based solution for single-image FB removal. To the best of our knowledge, this is the **first** work to formulate the task with neural networks. Since large-scale paired training data are unavailable, we first develop a ***simulation pipeline*** that synthesizes realistic FB in the luminance domain with stochastic variations in banding angle, spacing, and width, together with feathered transitions and sensor noise. For rigorous evaluation, we further collect a paired ***real-world dataset*** with pixel-aligned banded observations and banding-free references. Building on PiSA-SR [6], we then propose **R**emoval of **I**mage **F**licker-banding via **L**atent diffusion **E**nhancement (***RIFLE***), which introduces two task-specific components: (I) ***Banding-Suppressed High-Frequency Prior (BSHP)***, which provides structural guidance by combining fine-grained high-frequency responses with smooth local support, and (II) ***Masked Loss*** **(ML)**, which emphasizes banded regions while preserving global fidelity. Experiments on our real-world FB dataset show that RIFLE consistently outperforms recent image reconstruction baselines.

As shown in Fig. 1, our main contributions to single-image FB removal are as follows:

- To the best of our knowledge, it is ***the first work*** to research the simulation and removal of the flicker-banding (FB) and establishes a strong foundation for future research.
- We propose a FB simulation pipeline to build a large-scale training dataset and real-world testing dataset for evaluating the FB removal model's performance in practical settings.

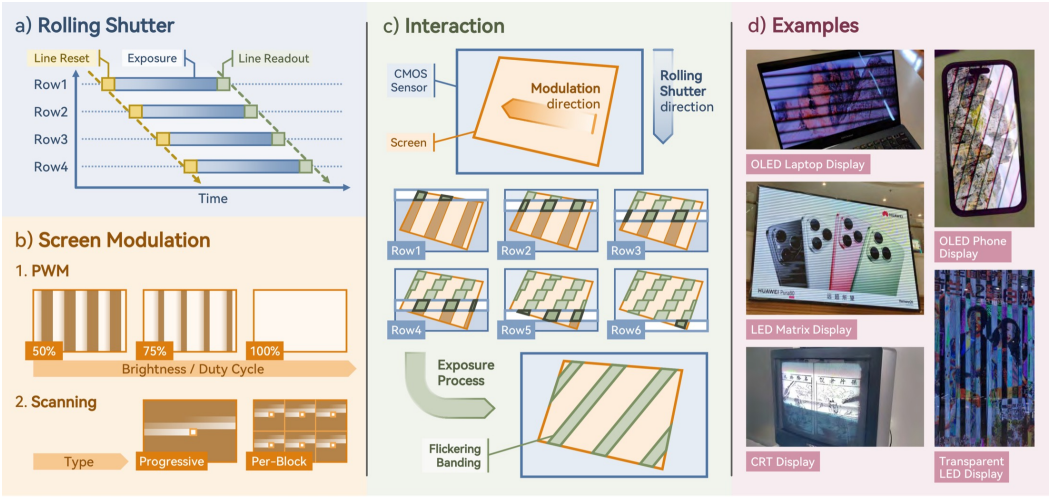


Figure 2: Flicker-banding (FB) when filming screens with smartphone cameras. **a)** Rolling shutter exposure process. **b)** Typical display brightness modulation (e.g., PWM and scanning refresh). **c)** Interaction between the camera exposure and screen modulation leads to the banding artifacts. **d)** Example banding patterns captured from different modern display technologies.

- We present ***RIFLE***, a one-step diffusion-based model with a Banding-Suppressed High-Frequency Prior (***BSHP***) that highlights informative high-frequency transitions for task-specific prior-guided FiLM modulation, and a Masked Loss (***ML***) tailored to FB artifacts.
- Our RIFLE achieves substantial improvement over other recent image reconstruction methods in both the quantitative metrics and visual comparisons. RIFLE is capable of addressing various kinds of FB patterns and can also be directly applied to real-world scenarios.

2 Related work

Researchers have addressed some display-camera artifacts, such as the moiréing patterns. Many works [7, 8, 2, 3] make great progress in constructing the moiréing datasets and designing the demoiréing models. Other research on the rolling-shutter degradation spans the physics-based and learning-based correction, including joint the rolling-shutter correction and deblurring [9, 10]. Parallel efforts on the flickering artifacts target fluctuations caused by temporal variations in global illumination such as fluorescent-light flicker [11, 12] using typical methods like CycleGAN [13]. More recently, BurstDeflicker [14] introduces a benchmark for burst deflickering in dynamic scenes under varying ambient illumination. FlickerFormer [15] further exploits periodicity and directionality cues across burst frames to improve ambient-light flicker removal.

However, the above flicker-removal studies address ambient-illumination fluctuations rather than the display-camera flicker-banding considered here, which arises from the interaction between rolling-shutter smartphone capture and PWM- or scan-driven screens. Compared with burst deflickering, our setting is more challenging because the banding is stronger, more complex, and not restricted to horizontally aligned stripes, making prior assumptions and temporal cues less effective. To the best of our knowledge, no public paired banding dataset is currently available in either academia or industry for this setting. Our work releases the first dataset for rigorous evaluation and formulates flicker-banding restoration based on diffusion models [16–18].

3 Preliminaries

Flicker-Banding (FB). When smartphone cameras capture OLED or LED-matrix displays, periodic bright and dark stripes may appear in the image. These FB artifacts arise from the temporal mismatch between rolling-shutter acquisition and display brightness modulation.

Most smartphone cameras use CMOS sensors with a rolling-shutter readout [19], which exposes and reads image rows sequentially and thus becomes sensitive to time-varying illumination.

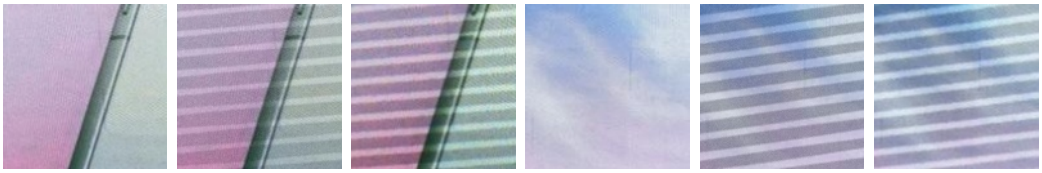


Figure 3: Visual comparison between our simulated flicker-banding and real-world flicker-banding. GT indicates the real-world non-banding images, on which our simulation pipeline is conducted. SIM indicates our simulation FB images, while LQ indicates real-world FB images.

OLED displays typically regulate brightness through pulse-width modulation (PWM) [20], while LED matrices often rely on scanning refresh. In both cases, only part of the screen is lit at a given moment, introducing high-frequency temporal fluctuations during image capture.

FB appears when sequential exposure overlaps with the screen's modulated emission [21], as illustrated in Fig. 2. This temporal aliasing converts otherwise invisible temporal dynamics into visible stripe patterns in the captured image, whose appearance depends on both camera settings and display modulation. Further details of these display mechanisms are provided in Appendix A.1.

Common Case: Gray Banding. Although extreme imaging conditions and exposure parameters can produce colored or curved stripes, everyday photography more often yields straight gray-scale bands, which are the dominant artifact in our dataset and the primary focus of our restoration setting.

Distinction from Moiré. Although both moiré and FB arise from display-camera interactions, moiré is caused by *spatial* interference between the display lattice and the camera sampling grid, whereas FB results from *temporal* aliasing between the rolling-shutter readout and display modulation. Their different origins make FB a distinct restoration problem.

4 Methods

4.1 Flicker-Banding Dataset Pipeline

To the best of our knowledge, no existing dataset targets flicker-banding (FB) degradations. We therefore build a simulation pipeline in Fig. 4 for training and a real-world dataset for testing.

4.1.1 Simulated Flicker-Banding Datasets for Training Datasets

We simulate training degradations because large-scale paired real FB data are scarce.

Let the high-quality (HQ) RGB image be $I_{HQ} \in [0,1]^{3\times H\times W}$ with pixel coordinates (x, y). We form a stripe-aligned local coordinate (u, v) by rotating pixel coordinates through banding angle θ:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x - \frac{W-1}{2} \\ y - \frac{H-1}{2} \end{bmatrix}, \theta \in [-\pi, \pi). \tag{1}$$

Given nominal stripe width $w > 0$ and gap $g > 0$, we get the spatial period $P = w + g$.

With a normal-direction phase offset ϕ (in pixels), the centerline of the k-th stripe is

$$L_k^c(u) = kP + \phi, \qquad k \in \mathbb{Z}. \tag{2}$$

The basic FB mask (1 indicates the banding area, 0 indicates the non-banding area) is

$$\mathcal{M}(u, v) = \begin{cases} 1, & \exists\, k : \ |v - L_k^c(u)| \le \frac{w_0}{2}, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

To better mimic real FB, we introduce jitter to stripe orientation, spacing, width, and boundaries.

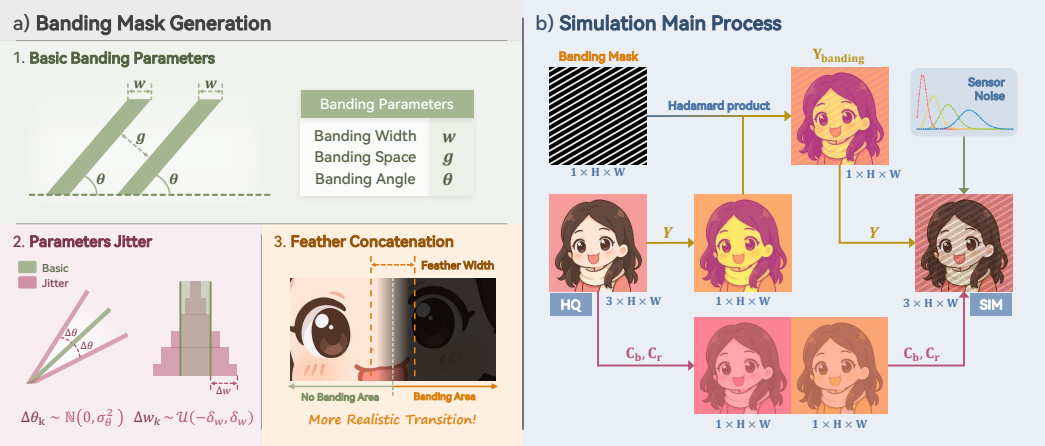


Figure 4: Our flicker-banding simulation pipeline design. **a)** Stage 1: We generate the general framework based on the basic banding parameters and introduce parameter jitter and feather concatenation for a more realistic transition. **b)** Stage 2: We overlay the flicker-banding mask on the Y-channel of high-quality (HQ) images and add sensor noise to the reconstructed screen images.

Orientation angle jitter. To allow realistic departures from the nominal orientation θ, we model the local orientation of the k-th stripe along its tangential coordinate u as

$$\theta_k(u) = \theta_0 + \Delta\theta_k(u), \tag{4}$$

where $\triangle\theta_k(u) \sim \mathcal{N}(0, {\sigma_\theta}^2)$ is a zero-mean Gaussian perturbation with the variance σ_θ^2.

Spacing and width fluctuation. We denote the k-th stripe centerline with spacing jitter as

$$L_k^{\mathrm{c}}(u) = kP + \phi + \Delta g_k, \qquad \Delta_g \sim \mathcal{U}(-\delta_g, \delta_g)\,, \tag{5}$$

where δ_g is the spacing-jitter amplitude. The banding width jitter with width-jitter amplitude δ_w is

$$w_k = w + \Delta w_k, \qquad \Delta_w \sim \mathcal{U}(-\delta_w, \delta_w)\,. \tag{6}$$

Axial edge fluctuation. The top and bottom edges meander along the stripe axis:

$$\begin{cases} v_k^{\mathrm{top}}(u) = v_k^{\mathrm{c}}(u) + \frac{1}{2} w_k(u) + \delta_{\mathrm{edge}}\, \eta_{\mathrm{top}}(u) \\ v_k^{\mathrm{bot}}(u) = v_k^{\mathrm{c}}(u) - \frac{1}{2} w_k(u) + \delta_{\mathrm{edge}}\, \eta_{\mathrm{bot}}(u) \end{cases}, \tag{7}$$

where $\eta_{\mathrm{top}}, \eta_{\mathrm{bot}}$ are low-pass 1D random processes and δ_{edge} sets the normal jitter amplitude.

We convert I_{HQ} from RGB to $\mathrm{YC_bC_r}$ and isolate the luminance channel:

$$I_{HQ}^Y = K_R I_{HQ}^R + K_G I_{HQ}^G + K_B I_{HQ}^B, \tag{8}$$

where $[K_R, K_B, K_G] = [0.299, 0.114, 0.587]$, and $I^C \in [0,1]^{1\times H\times W}$ denotes channel C of image I. Banding mask is then applied only on luminance channel with darkening factor $v_Y \in (0, 1]$:

$$I_{LQ}^Y = \underbrace{v_Y I_{HQ}^Y \mathcal{M}}_{Banding\ Area} + \underbrace{I_{HQ}^Y (1 - \mathcal{M})}_{Nonbanding\ Area}\,. \tag{9}$$

This luminance-only design is motivated by our observation that real FB mainly appears in the Y channel of $\mathrm{YC_bC_r}$, while the chrominance channels remain nearly unchanged. As further shown in Appendix A.6, swapping only the Y channel between banded and banding-free images transfers the artifact almost entirely. Applying the same mask to all RGB channels therefore introduces a noticeable domain gap in real-world screen photography.

Finally, we obtain the simulated FB image I_{LQ} by recombining the processed channels and adding sensor noise to better emulate real short-exposure capture conditions in practice:

$$I_{LQ} = \mathcal{C}(I_{LQ}^Y, I_{HQ}^{C_b}, I_{HQ}^{C_r}) + \zeta(\mathcal{C}(I_{LQ}^Y, I_{HQ}^{C_b}, I_{HQ}^{C_r})), \tag{10}$$

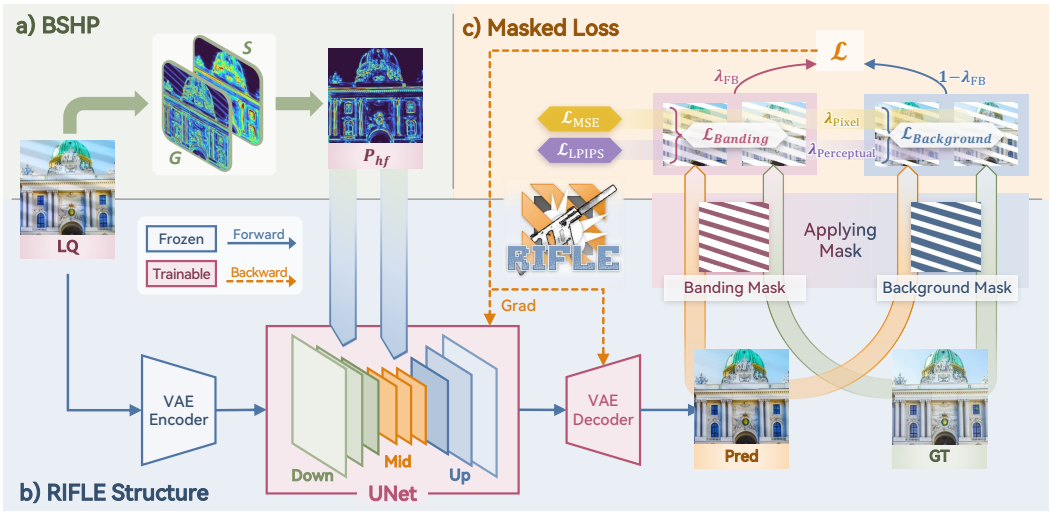


Figure 5: Overview of our model design. **a)** We propose Banding-Suppressed High-Frequency Prior (**BSHP**), which combines a fine-grained gradient signal with a smooth structural support map to build a compact single-channel prior. **b)** We adopt a one-step diffusion restoration model and inject BSHP through multi-stage FiLM modulation for degradation-aware feature recalibration. **c)** We propose a masked loss (**ML**) that emphasizes banded regions while preserving global fidelity.

Here, $\mathcal{C}$ converts $\mathrm{YC_bC_r}$ to RGB. The noise term ζ combines Poisson strength α and Gaussian strength σ_r^2 to mimic mixed noise that arises during short-exposure camera capture:

$$\zeta(I) = \sqrt{\alpha I + \sigma_r^2}\,\varepsilon, \qquad \varepsilon \sim \mathcal{N}(0,1). \tag{11}$$

We also apply feathered blending at stripe boundaries for smoother transitions. Fig. 3 shows that the simulated FB is visually close to real FB, supporting the validity of the training data.

4.1.2 Real-World Flicker-Banding Datasets for Testing Datasets

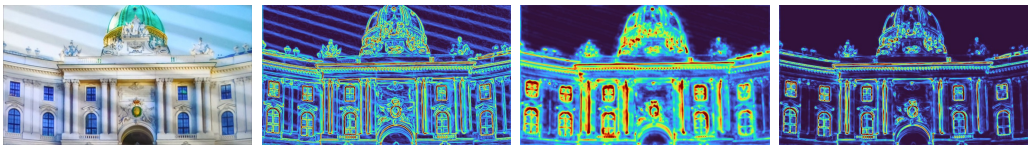
Although simulation provides abundant training samples, objective evaluation requires a real-world benchmark. We collect 105 valid high-resolution pairs from five scenes on commercial LED/OLED-type emissive advertising displays, using a vivo X200s under indoor ambient lighting. Each pair is captured from a fixed viewpoint: a short exposure produces the banded LQ observation, while a longer exposure produces the banding-free GT reference, with ambient lighting unchanged during paired capture. The scene-specific exposure and ISO settings are reported in Appendix A.9.

We register each LQ image to its GT reference using a homography estimated from matched image features, with ECC refinement when feature-based alignment is insufficient. We crop the common valid field of view and visually discard pairs with registration failures or invalid boundaries. Paired patches are then extracted at identical coordinates; a screen mask selects locations containing sufficient display content without modifying or replacing any LQ or GT pixels. The dataset comprises 105 image pairs at 4096×3072 resolution, from which we obtain 424 paired 512×512 test patches.

Our intended scope is dark, predominantly luminance, approximately straight FB with arbitrary orientations and varying width, spacing, and strength. The acquisition conditions instantiate this predefined target. Chromatic, strongly curved or nonstationary, multi-frequency or per-block patterns, and severe cases with unrecoverable information loss remain beyond the current capability. We do not claim coverage of CRTs, projectors, fluorescent lamps, ambient LED light sources, or other flicker sources.

4.2 Design of Flicker-Banding Removing Model

Although FB is common in screen photography, effective solutions remain limited. Hardware-side correction is difficult because of diverse screen scanning mechanisms and material properties. We therefore adopt the restoration model in Fig. 5 and use the one-step diffusion image restoration model PiSA-SR [6] as the baseline for its strong efficiency and performance.



LQ | Gradient signal G | Repeatability map S | Banding cue P_{hf}

Figure 6: Visualization of BSHP. Gradient signal G captures fine transitions but also stripe responses, while repeatability map S is smoother and provides structural support. Their fusion yields a compact prior P_{hf} that preserves informative transitions while suppressing unstable banding responses.

The contribution of BSHP and ML lies in their task-specific integration into the complete FB-removal pipeline rather than in FiLM or masked losses themselves. BSHP fuses gradient-based localization with repeatability-based structural support and injects this FB-specific prior through multi-stage FiLM. ML uses the banding mask produced by the same simulation process, coupling degradation synthesis with region-focused supervision for banded image regions.

4.2.1 Loss Design of Removing Model

Let the low-quality (LQ), prediction, and ground truth (GT) images be x_{LQ}, x_{Pred}, and $x_{\mathrm{GT}} \in \mathbb{R}^{B\times C\times H\times W}$, and let the simulation mask be $\mathcal{M} \in \{0,1\}^{B\times 1\times H\times W}$, shared with the simulated FB generation process, where 1 indicates banding and 0 indicates background. Since the background of x_{LQ} is typically close to that of x_{GT}, we emphasize reconstruction in the banding region using $\lambda_{\mathrm{FB}} \in [0,1]$, with $\lambda_{\mathrm{Pixel}}, \lambda_{\mathrm{Perceptual}} \geq 0$ weighting the loss terms.

Area-decoupled masked mean operator. Let $I \in \mathbb{R}^{B\times C\times H\times W}$ be an image tensor and $\mathcal{M} \in \{0,1\}^{B\times 1\times H\times W}$ the binary banding mask. Let $\tilde{\mathcal{M}}$ be $\mathcal{M}$ broadcast to the shape of I. We define

$$\langle\!\langle I \rangle\!\rangle_{\mathcal{M}} = \frac{\sum I \odot \tilde{\mathcal{M}}}{\sum_{b,c,h,w} \mathbf{1} \odot \tilde{\mathcal{M}} + \varepsilon}, \tag{12}$$

where $\varepsilon > 0$ ensures numerical stability for the area-normalized pixel loss in each region.

Masked Pixel Loss. We use mean squared error (MSE) for pixel-level reconstruction:

$$\mathcal{L}_{\mathrm{Pixel}} = (x_{\mathrm{Pred}} - x_{\mathrm{GT}})^2, \tag{13}$$

To emphasize restoration in banding regions, we apply $\mathcal{M}$ to obtain the masked MSE loss:

$$\mathcal{L}_{\mathrm{Pixel}}^{\mathrm{Masked}} = \lambda_{\mathrm{FB}} \langle\!\langle \mathcal{L}_{\mathrm{Pixel}} \rangle\!\rangle_{\mathcal{M}} + (1-\lambda_{\mathrm{FB}}) \langle\!\langle \mathcal{L}_{\mathrm{Pixel}} \rangle\!\rangle_{\mathbf{1}-\mathcal{M}}. \tag{14}$$

Masked Perceptual Loss. We use LPIPS [22] to compare masked image pairs. Let Φ denote the scalar LPIPS distance, including its required input normalization. For $\mathcal{R} \in \{\mathcal{M}, \mathbf{1}-\mathcal{M}\}$, we apply the same mask to both input images before computing the LPIPS distance:

$$d_\Phi(\mathcal{R}) = \frac{1}{B}\sum_{b=1}^{B} \Phi\Big(\widetilde{\mathcal{R}}_b \odot x_{\mathrm{Pred},b}, \widetilde{\mathcal{R}}_b \odot x_{\mathrm{GT},b}\Big). \tag{15}$$

Here, $\widetilde{\mathcal{R}}_b$ broadcasts the mask across channels; both inputs are zero outside the selected region. We combine the two batch-averaged scalar distances using the banding weight:

$$\mathcal{L}_{\mathrm{Perceptual}}^{\mathrm{Masked}} = \lambda_{\mathrm{FB}} d_\Phi(\mathcal{M}) + (1-\lambda_{\mathrm{FB}}) d_\Phi(\mathbf{1}-\mathcal{M}). \tag{16}$$

Merged Masked Loss. The final loss balances pixel fidelity and overall perceptual quality:

$$\mathcal{L} = \lambda_{\mathrm{Pixel}} \mathcal{L}_{\mathrm{Pixel}}^{\mathrm{Masked}} + \lambda_{\mathrm{Perceptual}} \mathcal{L}_{\mathrm{Perceptual}}^{\mathrm{Masked}}. \tag{17}$$

4.2.2 Banding-Suppressed High-Frequency Prior (BSHP)

Inspired by conditional restoration, we cast FB removal as a prior-guided restoration problem. Rather than explicitly estimating a stripe mask, we seek a compact prior that highlights recoverable high-frequency structures while suppressing unstable artifact responses.

For an input image x_{LQ}, we first convert its RGB channels to luminance for prior construction:

$$Y = K_R R + K_G G + K_B B. \tag{18}$$

To characterize informative local high-frequency structures in the degraded input, we apply mild Gaussian smoothing before computing the spatially localized gradient signal:

$$\tilde{Y} = \mathcal{G}_{\sigma_b}(Y), \qquad G = \mathrm{Norm}\left(\sqrt{|\partial_x \tilde{Y}|^2 + |\partial_y \tilde{Y}|^2}\right). \tag{19}$$

Here, G localizes candidate high-frequency structures, but also responds to FB-induced stripe edges.

However, transition strength alone is insufficient for robust prior construction. We therefore introduce a second term that provides smoother and more reliable local structural support:

$$R_{\mathrm{mid}} = \alpha\big(Y - \mathcal{G}_{\sigma_1}(Y)\big) + (1-\alpha)\big(Y - \mathcal{G}_{\sigma_2}(Y)\big), \qquad X = \mathrm{Norm}(|R_{\mathrm{mid}}|), \tag{20}$$

where $\sigma_1 < \sigma_2$ and $\alpha \in [0,1]$. We then define a repeatability map by searching over candidate directions Θ and spatial offsets Δ across the local neighborhood of each pixel:

$$S(x,y) = \mathrm{Norm}\left(\max_{\delta\in\Delta,\ \theta\in\Theta} \mathcal{G}_{\sigma_r}\big(X(x,y)\cdot X\big((x,y)+\delta u_\theta\big)\big)\right), \tag{21}$$

where u_θ is the unit vector of direction θ. We use $\Theta = \{0^\circ, 15^\circ, \ldots, 165^\circ\}$, $\Delta = \{2, 4, \ldots, 20\}$ pixels, and $\sigma_r = 2.0$. Here, S is a local repeatability score, not an estimator of stripe frequency or angle: it measures the strength of similar luminance responses within the local neighborhood. Spatial band frequency and angle affect the direction-offset responses before aggregation, but the maximum operation retains repetition strength rather than a particular direction or offset. Thus, θ and δ cannot be compared one-to-one with the angle, period, phase, width, or mask used in simulation. The simulation parameters generate training degradations, whereas S is computed directly from each LQ image. When S reliably supports relevant structures, its fusion with the localized gradient cue G provides more informative guidance for FB restoration in each image region.

Neither cue alone is sufficient: G is rich in fine details but also contains stripe responses, while S is smoother and cleaner but lacks precise localization. We therefore fuse these two complementary cues multiplicatively at each pixel and define the FB-specific final prior as

$$P_{\mathrm{hf}} = \mathrm{Norm}(G^{\gamma_g} \cdot S^{\gamma_s}), \tag{22}$$

where γ_g and γ_s balance transition strength and structural support. The gradient term preserves precise localization, while the support term reweights these responses with smoother local context. As shown in Fig. 6, P_{hf} is a single-channel prior that retains the precise localization ability of G while suppressing unstable stripe-dominated responses through the structural support map S.

Rather than concatenating the prior to the input image, we inject it into the restoration backbone through FiLM-based feature modulation. We first encode the prior as

$$M = \mathrm{SiLU}(\mathrm{Conv}_{1\times1}(P_{\mathrm{hf}})), \tag{23}$$

$$z = [\mathrm{Pool}_g(M); \mathrm{Pool}_h(M); \mathrm{Pool}_v(M)], h = \mathrm{SiLU}(W_s z + b_s), \tag{24}$$

and, for each injected U-Net stage l, generate channel-wise modulation parameters:

$$[\gamma_l, \beta_l] = W_l h + b_l, \qquad \hat{F}_l = F_l \odot (1+\gamma_l) + \beta_l. \tag{25}$$

Here, F_l denotes the intermediate feature tensor at stage l, while Pool_g, Pool_h, and Pool_v denote global, horizontal, and vertical average pooling. This directional FiLM conditioning biases the backbone toward structurally informative high-frequency transitions with minimal overhead.

5 Experiments

5.1 Experiments Setup

Data Construction. For training, we apply our simulation pipeline to LSDIR [26] and UHDM [8]. LSDIR is a large-scale super-resolution dataset, while UHDM is a demoiréing dataset; both provide abundant high-quality images for synthesizing flicker-banding degradations. We use UHDM to better cover screen-shooting scenarios and LSDIR to improve generalization to more diverse natural images. For the testing datasets, we use our proposed real-world datasets, consisting of 424 paired patches of 512×512 size. Results from real-world datasets better demonstrate the model's practical value.

Table 1: Quantitative comparison on the real-world FB testing dataset. Except for Notch Filter and FlickerFormer*, all learning-based baselines are trained or finetuned on our simulated data. FlickerFormer* uses the official checkpoint from the original deflickering task, while FlickerFormer denotes the version finetuned on our simulated data. Best results are highlighted in red.

Methods	Task	SSIM↑	ms-SSIM↑	LPIPS↓	DISTS↓	FSIM↑	GMSD↓
LQ	–	0.6239	0.6758	0.3070	0.1978	0.7947	0.2045
Notch Filter	Filtering	0.5562	0.6313	0.3411	0.2600	0.7196	0.2110
MAT [23]	Super-resolution	0.4747	0.6479	0.4114	0.2243	0.7766	0.2003
InvSR [24]	Super-resolution	0.6101	0.6873	0.3833	0.2195	0.8035	0.2108
Step1X [25]	Image Editing	0.6164	0.6715	0.3564	0.2066	0.7900	0.2079
ESDNet [8]	Demoiréing	0.6306	0.6941	0.2984	0.1905	0.8107	0.1969
FlickerFormer* [15]	Deflickering	0.6252	0.6783	0.3054	0.1968	0.7977	0.2031
FlickerFormer [15]	Deflickering	0.6658	0.7993	0.2433	0.1534	0.9086	0.1528
PiSA-SR [6]	Super-resolution	0.6201	0.7370	0.3808	0.2048	0.8443	0.2085
RIFLE (ours)	Debanding	0.6843	0.8318	0.2209	0.1410	0.9281	0.1376

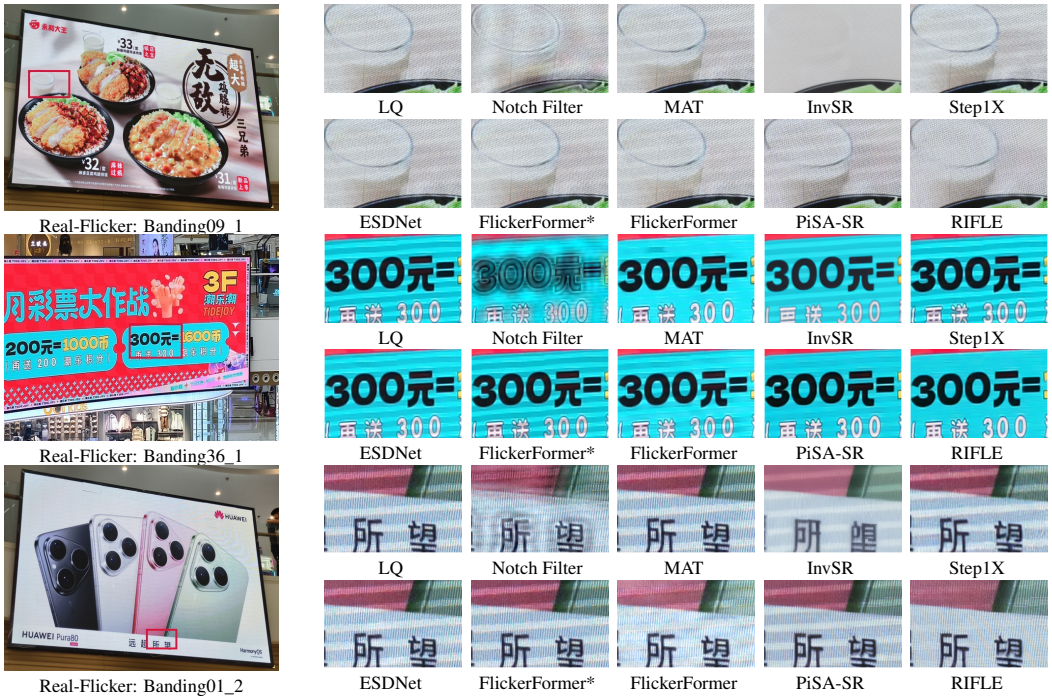


Figure 7: Visual comparison on three representative Real-Flicker cases. RIFLE removes stronger flicker-banding artifacts with cleaner structures and fewer residual stripes than competing baselines.

Evaluation Metrics. No metric directly quantifies FB severity. Since bands affect limited areas while preserving detail elsewhere, global scores can miss visible improvements. We therefore complement a user study with SSIM [27], LPIPS [22], DISTS [28], FSIM [29], and GMSD [30]. We omit non-reference metrics because they score banded and banding-free images similarly.

Implementation Details. We finetune rank-32 LoRA for 50,000 iterations on paired 512×512 images (batch size 4, learning rate 5×10^{-5}), using about 43.3 GB of GPU memory and 25.8 hours. The masked-loss weights are $\lambda_{\text{FB}} = 0.8$, $\lambda_{\text{Pixel}} = 1.0$, and $\lambda_{\text{Perceptual}} = 2.0$.

Compared Methods. With no prior FB-specific model available, we finetune related baselines on our simulated data: MAT [23] and Step1X [25] (transformer/image editing), InvSR [24] and PiSA-SR [6] (diffusion), ESDNet [5] (demoiréing), and FlickerFormer [15] (deflickering).

5.2 Main Results

Quantitative Results. Tab. 1 reports some quantitative results on our real-world dataset. RIFLE outperforms other competing methods on most metrics. Notably, even raw FB inputs achieve relatively high scores, as reference-based metrics are largely insensitive to thr banding artifacts—FB entails minimal fine-detail loss and non-banding regions closely resemble the corresponding GT reference.

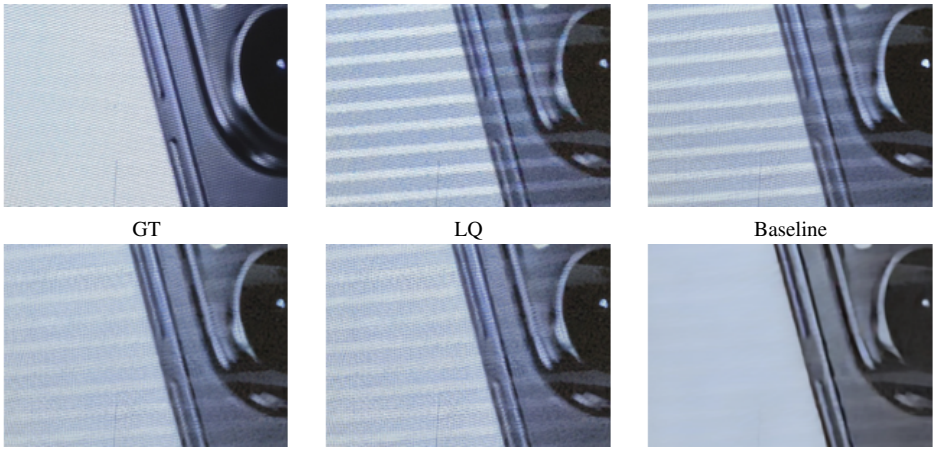


Figure 8: Visual comparison of RIFLE component ablations on a real-world FB image.

Table 2: Ablation study of our proposed components on real-world FB datasets.

Methods	SSIM↑	ms-SSIM↑	LPIPS↓	DISTS↓	FSIM↑	GMSD↓
Baseline	0.6201	0.7370	0.3808	0.2048	0.8443	0.2085
ML	0.6673	0.7899	0.2520	0.1621	0.8884	0.1513
BSHP	0.6668	0.7928	0.2521	0.1587	0.8920	0.1528
ML+BSHP	0.6843	0.8318	0.2209	0.1410	0.9281	0.1376

Visual Comparison. Fig. 7 compares real-world results. Competing methods leave visible FB residues despite finetuning on our simulated data. FlickerFormer mainly targets horizontal stripes in its official model; finetuning helps, but stripes and content distortion remain. RIFLE nearly removes mild banding with high fidelity and removes most severe banding with only minor residuals in the output, while other compared methods improve little.

Computational Cost and Additional Evaluation. At 512×512 resolution on an RTX A6000, RIFLE adds 1.239 M parameters, 0.0013 GiB peak inference memory, and 38.2 ms latency (13.7%) relative to PiSA-SR, with nearly unchanged profiled MACs/FLOPs. Appendix A.10 reports training and inference costs. The user-study protocol and statistics are detailed in Appendix A.4; OCR-based text fidelity and full-dataset video evaluation are reported in Appendices A.11 and A.12.

5.3 Ablation Study

Tab. 2 and Fig. 8 ablate masked loss (ML) and banding-suppressed high-frequency prior (BSHP). The baseline leaves stripes; ML targets banded regions for cleaner, more faithful outputs. BSHP adds localized high-frequency guidance with smooth structural support, improving removal but sometimes softening texture alone. Together, they best remove bands while preserving edges and detail across all evaluated metrics in our study, especially under heavier FB degradation.

6 Conclusion

We study flicker-banding (FB) in display photographs and introduce RIFLE with a simulation pipeline and paired real-world benchmark. Its compact prior, FiLM guidance, and region-focused masked loss remove bands while preserving detail. RIFLE outperforms recent baselines quantitatively and visually; ablations confirm both components. To our knowledge, this is the first neural study of FB removal with practical training-data synthesis and paired real-world evaluation data construction. Dark luminance bands in LED/OLED captures define the evaluated scope; chromatic, curved, and extreme information-loss patterns remain challenging. OCR results further show that character-level fidelity is not guaranteed. These limitations motivate richer simulations and text-preserving restoration.

Acknowledgments and Disclosure of Funding

This work was supported by Huawei Explore X funds. This work was also sponsored by Al Hundred Schools Program. The authors declare no competing financial interests related to this work.

References

[1] Yuxin Zhang, Mingbao Lin, Xunchao Li, Han Liu, Guozhi Wang, Fei Chao, Shuai Ren, Yafei Wen, Xiaoxin Chen, and Rongrong Ji. Real-time image demoireing on mobile devices. In *ICLR*, 2023.

[2] Shuning Xu, Binbin Song, Xiangyu Chen, and Jiantao Zhou. Image demoireing in raw and srgb domains. In *ECCV*, 2024.

[3] Xiaoyang Liu, Bolin Qiu, Jiezhang Cao, Zheng Chen, Yulun Zhang, and Xiaokang Yang. Freqformer: Image-demoiréing transformer via efficient frequency decomposition. *arXiv preprint arXiv:2505.19120*, 2025.

[4] Yanting Mei, Zhilu Zhang, Xiaohe Wu, and Wangmeng Zuo. Image demoiréing using dual camera fusion on mobile phones. In *ICME*, 2025.

[5] Qirui Yang, Fangpu Zhang, Yeying Jin, Qihua Cheng, Pengtao Jiang, Huanjing Yue, and Jingyu Yang. Dsdnet: Raw domain demoiréing via dual color-space synergy. In *ICML*, 2025.

[6] Lingchen Sun, Rongyuan Wu, Zhiyuan Ma, Shuaizheng Liu, Qiaosi Yi, and Lei Zhang. Pixel-level and semantic-level adjustable super-resolution: A dual-lora approach. In *CVPR*, 2025.

[7] Yujing Sun, Yizhou Yu, and Wenping Wang. Moiré photo restoration using multiresolution convolutional neural networks. *TIP*, 2018.

[8] Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Jiajun Shen, Jia Li, and Xiaojuan Qi. Towards efficient and scale-robust ultra-high-definition image demoiréing. In *ECCV*, 2022.

[9] Zhihang Zhong, Yinqiang Zheng, and Imari Sato. Towards rolling shutter correction and deblurring in dynamic scenes. In *CVPR*, 2021.

[10] Mingdeng Cao, Sidi Yang, Yujiu Yang, and Yinqiang Zheng. Rolling shutter correction with intermediate distortion flow estimation. *arXiv preprint arXiv:2404.06350*, 2024.

[11] Sebastian Köhler, Giulio Lovisotto, Simon Birnbach, Richard Baker, and Ivan Martinovic. They see me rollin': Inherent vulnerability of the rolling shutter in cmos image sensors. *arXiv preprint arXiv:2101.10011*, 2021.

[12] Xiaodan Lin, Yangfu Li, Jianqing Zhu, and Huanqiang Zeng. Deflickercyclegan: Learning to detect and remove flickers in a single image. *TIP*, 2023.

[13] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[14] Lishen Qu, Zhihao Liu, Shihao Zhou, Yaqi Luo, Jie Liang, Hui Zeng, Lei Zhang, and Jufeng Yang. Burstdeflicker: A benchmark dataset for flicker removal in dynamic scenes. In *NIPS*, 2025.

[15] Lishen Qu, Shihao Zhou, Jie Liang, Hui Zeng, Lei Zhang, and Jufeng Yang. It takes two: A duet of periodicity and directionality for burst flicker removal. In *CVPR*, 2026.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[17] Bin Xia, Yulun Zhang, Shiyin Wang, Yitong Wang, Xinglong Wu, Yapeng Tian, Wenming Yang, and Luc Van Gool. Diffir: Efficient diffusion model for image restoration. *arXiv preprint arXiv:2303.09472*, 2023.

[18] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[19] Daniel Durini. *High performance silicon imaging: Fundamentals and applications of CMOS and CCD sensors*. Woodhead Publishing, 2019.

[20] Bernard Geffroy, Philippe Le Roy, and Christophe Prat. Organic light-emitting diode (oled) technology: materials, devices and display technologies. *Polymer International*, 2006.

[21] Robert C. Sumner. Describing and sampling the led flicker signal. In *EI*, 2020.

[22] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[23] Chengxing Xie, Xiaoming Zhang, Linze Li, Yuqian Fu, Biao Gong, Tianrui Li, and Kai Zhang. Mat: Multi-range attention transformer for efficient image super-resolution. *TCSVT*, 2025.

[24] Zongsheng Yue, Liao Kang, and Chen Change Loy. Arbitrary-steps image super-resolution via diffusion inversion. In *CVPR*, 2025.

[25] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai, Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, Yibo Zhu, Binxing Jiao, Xiangyu Zhang, Gang Yu, and Daxin Jiang. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

[26] Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, et al. Lsdir: A large scale dataset for image restoration. In *CVPRW*, 2023.

[27] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.

[28] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *TPAMI*, 2020.

[29] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: a feature similarity index for image quality assessment. *TIP*, 2011.

[30] Wufeng Xue, Lei Zhang, Xuanqin Mou, and Alan C. Bovik. Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. *TIP*, 2014.

[31] Shuai Han, Imari Sato, Takahiro Okabe, and Yoichi Sato. Fast spectral reflectance recovery using dlp projector. *IJCV*, 2014.

[32] Qiming Hu, Linlong Fan, Yiyan Luo, Yuhang Yu, Xiaojie Guo, and Qingnan Fan. Text-aware real-world image super-resolution via diffusion model with joint segmentation decoders. In *NeurIPS*, 2025.

[33] Zheng Chen, Ruofan Yang, Jin Han, Dehua Song, Zichen Zou, Chunming He, Yong Guo, and Yulun Zhang. Diffst: Spatiotemporal-aware diffusion for real-world space-time video super-resolution. *arXiv preprint arXiv:2605.13182*, 2026.

[34] Zheng Chen, Bowen Chai, Rongjun Gao, Mingtao Nie, Xi Li, Bingnan Duan, Jianping Fang, Xiaohong Liu, Linghe Kong, and Yulun Zhang. Dvface: Spatio-temporal dual-prior diffusion for video face restoration. *arXiv preprint arXiv:2604.14560*, 2026.

A Technical appendices and supplementary material

A.1 Flicker-Banding Patterns across Display Technologies

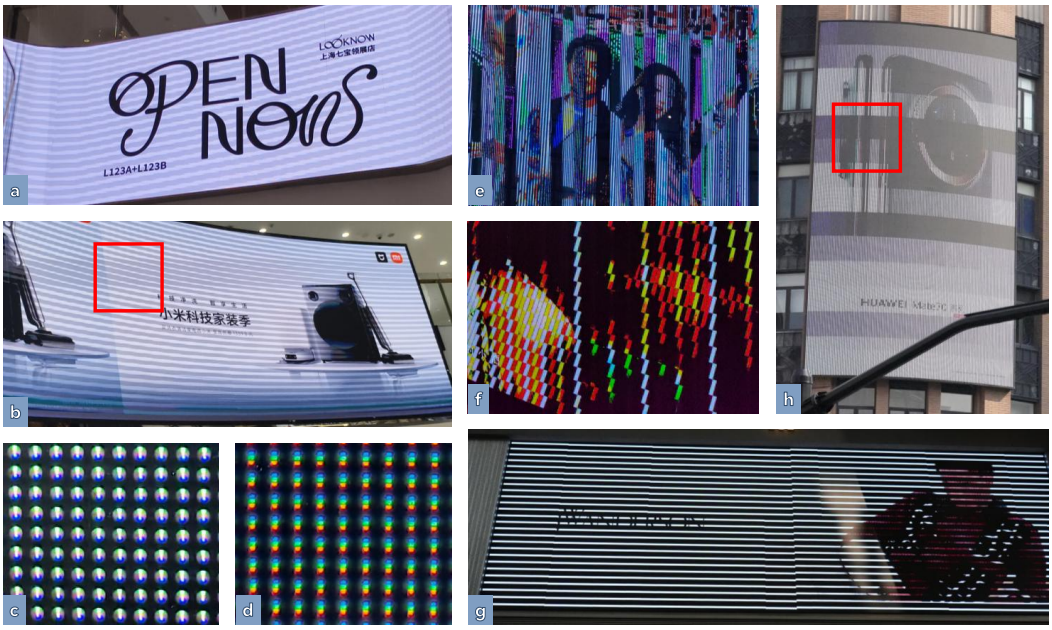


Figure 9: Flicker-banding patterns on LED matrix displays with different scanning refresh mechanisms captured under various conditions. **b)** Thickening of dark stripes, likely due to changes in PWM duty cycle. **c-d)** Different LED subpixel arrangements. **e-f)** Atypical banding patterns. **h)** Complex banding patterns caused by per-block scanning refresh.

This subsection surveys LED, OLED, CRT, and projector mechanisms to explain the broader origins of screen-capture banding; it does not imply that RIFLE covers all four technologies. RIFLE is designed and validated only for dark FB in OLED and LED display captures. The surveyed chromatic, complex, per-block, and other atypical patterns illustrate mechanisms beyond the validated scope. FB from CRTs, projectors, fluorescent lamps, ambient LED light sources, and other scenarios is outside the current scope. We distinguish this mechanism survey from the real-world benchmark acquisition and supported FB types described in Sec. 4 and Appendix A.9.

A.1.1 LED Matrix Displays

LED matrix displays often use a scanning refresh scheme, where rows or columns of pixels are activated in succession. The matrix is often refreshed in blocks or chunks of rows, leading to more complex banding patterns when filmed. PWM is also commonly used in LED matrices, further contributing to more complex banding effects.

A.1.2 OLED Displays

OLED displays typically use a combination of PWM and current modulation to achieve smooth brightness control [20]. In most cases, the duty cycle is same across the entire panel at a given brightness level, resulting in uniform banding patterns when filmed with a camera. However, some OLED panels may reduce the duty cycle in very dark scenes to enhance contrast, leading to non-uniform banding artifacts. Color shifts may also occur on edges of banding stripes due to phase differences in RGB subpixel driving.

A.1.3 CRT Displays

In cathode-ray tube (CRT) monitors, stripe artifacts are primarily related to the scanning process of the electron beam. The resulting banding patterns differ from those in OLED and LED displays,

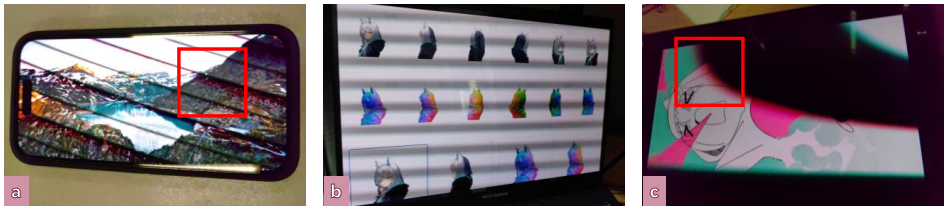


Figure 10: Flicker-banding patterns on OLED displays with PWM driving. **a)** Non-uniform banding in darker areas on an iPhone OLED screen. **b)** Horizontal gray banding on laptop OLED. **c)** Extremely bending and color shifts on banding edges.

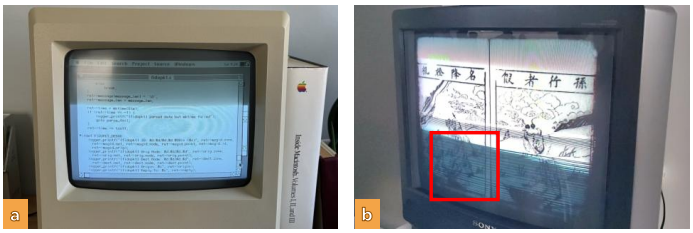


Figure 11: Flicker-banding patterns on CRT monitors with electron beam scanning. **b)** Phosphor persistence effect causing trailing artifacts, leaving a ghosting effect.

as factors including different scanning methods (e.g., interlaced vs. progressive) and phosphor persistence characteristics come into play.

A.1.4 Projectors

Projectors, especially those using digital light processing (DLP), can also exhibit banding artifacts when captured on camera [31]. These systems often employ spinning color wheels to generate full-color images, which can interact with the camera's rolling shutter to produce unique colored banding patterns. Brightness modulation is achieved through rapid micromirror switching, which can introduce non-uniform banding under certain filming conditions.

Across the surveyed display families, flicker-banding emerges from different timing mechanisms: row/column or per-block scanning and PWM dimming on LED matrix panels, a mixture of PWM and current-driven modulation with subpixel-phase offsets on OLEDs, raster scanning with phosphor persistence on CRTs, and time-sequential illumination (e.g., color wheels) in projectors. These mechanisms create bands that can be straight or curved (due to rolling-shutter readout), globally periodic or piecewise-structured (under per-block scanning/local modulation), and purely luminance-based or chromatic (due to subpixel phase shifts or color wheels).

Such diversity makes restoration challenging: the banding is frequently non-stationary, its frequency and pattern are strongly device-specific, and its appearance can vary with scene content, exposure settings, and viewing angle. Additionally, ISP processing and sensor noise further blur the boundary between true scene detail and artifact. This calls for future work to develop more advanced, physics-informed models and datasets that reflect this variety.

A.2 Additional Visual Comparison

We provide additional visual comparison for different methods on our Real-Flicker dataset in Fig. 14. More results demonstrate our RIFLE's outstanding performance compared to recent image reconstruction methods and show strong application potential in real-world scenarios.

A.3 Demoiré model performance in the case of Flicker-banding

In addition to the flikcer-banding (FB), moiré patterns are also bring great damage to the image quality when shooting screen. We find that the performance of demoiré model largely decrease when

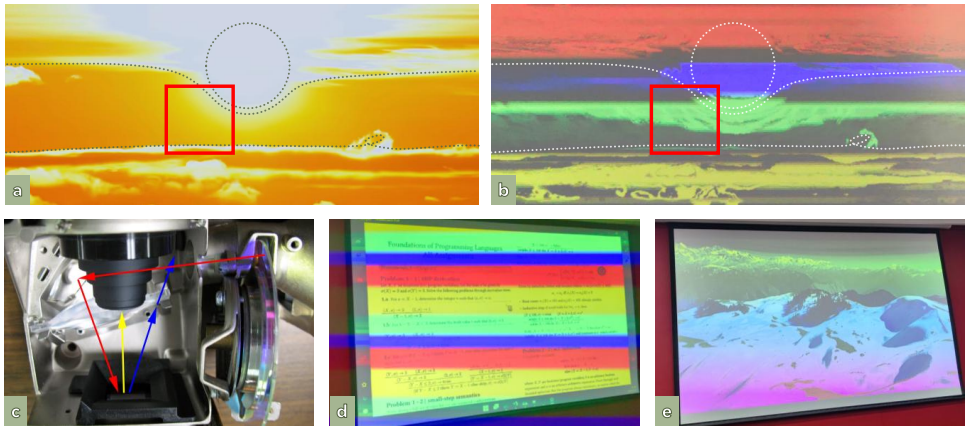
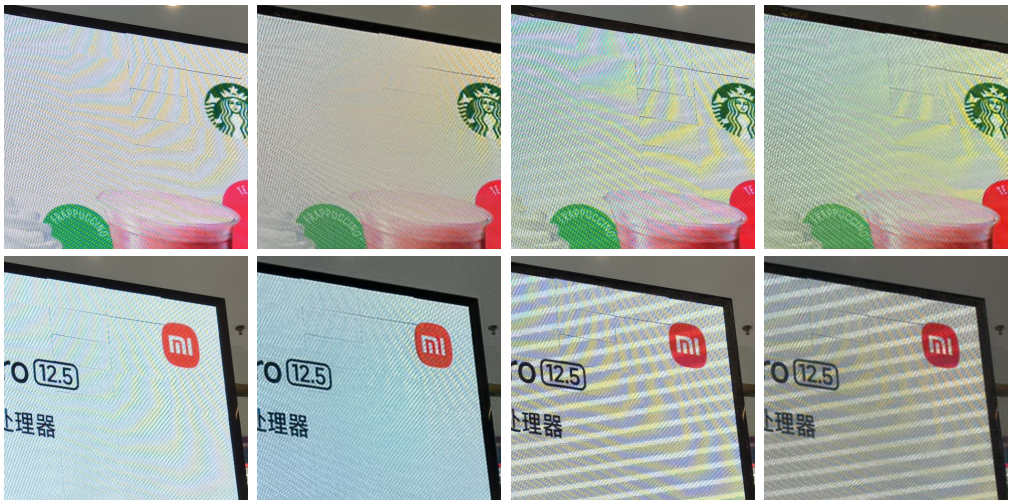


Figure 12: Flicker-banding patterns on DLP projectors with color wheel and micromirror switching. **a-b)** Comparison between original and projected images, highlighting the width of color stripes changes with brightness levels. **c)** Internal components adapted from DMahalko (CC BY-SA 3.0).



No FB before demoiré | No FB after demoiré | FB before demoiré | FB after demoiré

Figure 13: The demoiré performance comparison between no FB scene and FB scene.

both of the FB and moiré patterns exist, as shown in Fig. 13. It indicates the difference between FB and moiré patterns and removing them simultaneously is also a valuable issue to be solved.

A.4 User Study

Standard image-quality metrics do not directly measure FB severity, so we complement them with a single-choice preference study. Figure 15 shows the evaluation interface. All participants used the same interface under consistent viewing conditions; each trial displayed the eight outputs for the same test case simultaneously at the same size and layout, with method names hidden. Before the study, participants received the same explanation of the evaluation objective and selection criterion, together with practice guidance on operating the interface.

Each of the 50 participants completed one session of 100 trials, selecting the result with the best FB removal performance before proceeding to the next case. The study therefore produced 5,000 valid decisions across 100 test cases and eight methods. Scores in Table 3 are the number of selections for each method divided by the total number of valid decisions; they are preference shares, not rankings or numerical quality ratings.



Figure 14: Additional visual comparison on the remaining Section-1 cases that are not used in Fig. 7.

RIFLE received 4,120 selections, corresponding to 82.4%. At the decision level, the 95% Wilson confidence interval is [81.32%, 83.43%], and a one-sided exact binomial test against chance selection among eight methods ($p_0 = 1/8$) gives $p < 0.001$. These statistics summarize the observed decisions; they do not model dependence across repeated decisions from the same participant or test case.

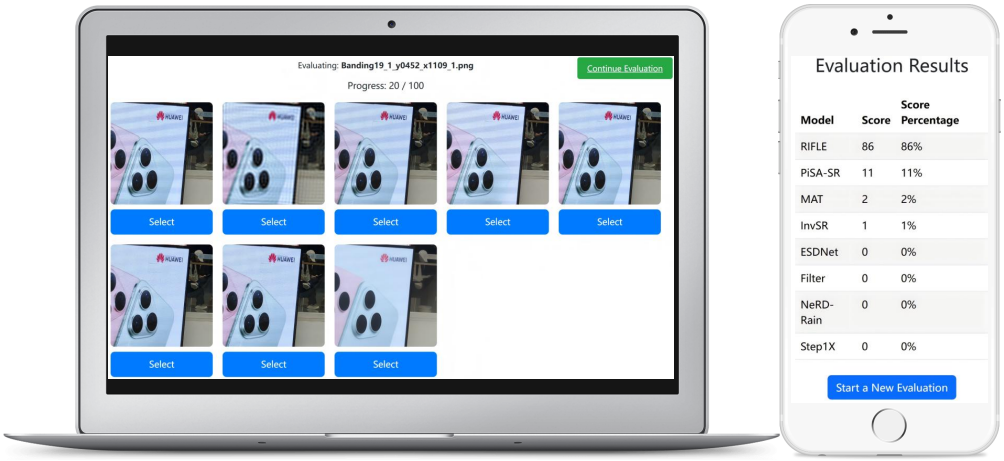


Figure 15: The operating system of the User Study.

Table 3: The results of User Study.

Methods	RIFLE	PiSA-SR	MAT	InvSR	Step1X	ESDNet	FlickerFormer	Notch Filter
Score/(%)	82.4	14.7	2.7	0.02	0	0.1	0.08	0

A.5 The out-of-distribution test results.

We provide the test results on other severe flicker-banding (FB) scenes in Fig. 16, not only on the dark straight FB scenes. The first row indicates that the black stripes are hard to recover and our model only played a minor role in filling in the gaps. The second row indicates that our model makes no difference to the stripes with severe color shifts. The third row implies that there is still a long way to go for our model to reconstruct an image whose details are completely destroyed.

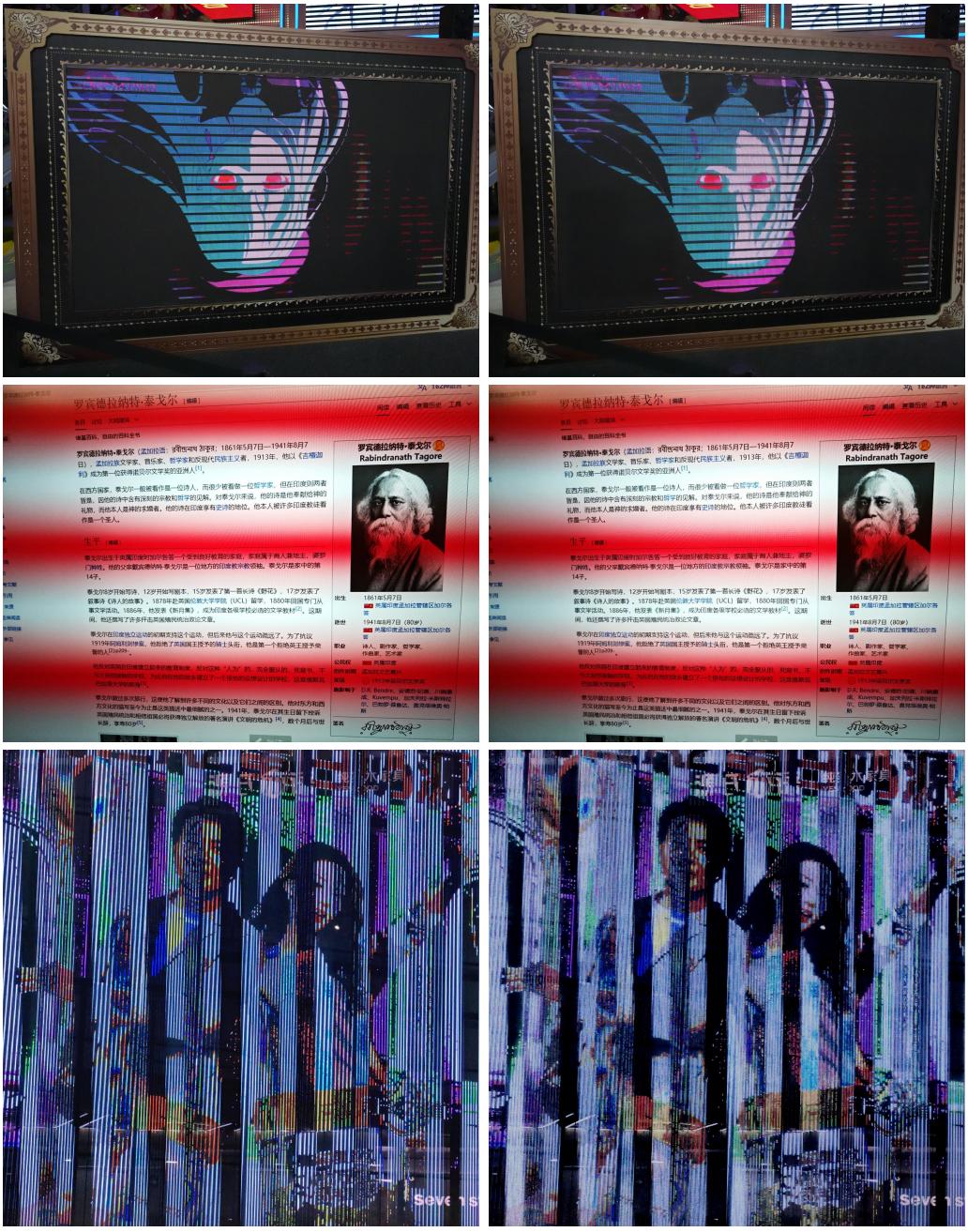
A.6 Analysis on the Flicker-banding simulation

Figures 17 and 18 compare GT and LQ images in RGB and YCbCr. Real FB is visible in all three RGB channels, but this does not imply that an identical stripe mask should be applied independently to R, G, and B. Under RGB–YCbCr conversion, modifying Y while preserving Cb/Cr changes all three reconstructed RGB channels. The luminance relation constrains the effective RGB masks to be channel-dependent rather than identical. Thus, the RGB observations and our Y-only simulation are consistent.

For the dark FB considered here, the chrominance channels Cb and Cr are much closer to GT, while Y exhibits pronounced banding. We therefore apply the banding mask to Y. The channel comparison supports this choice, and the simulation comparison below shows that applying the same mask independently to RGB yields no measurable improvement on real-world data, whereas Y-only simulation improves restoration. This evidence concerns the supported dark/luminance FB and does not establish coverage of chromatic banding.

To further validate this observation, we swapped the Y channels between GT and LQ while keeping their respective Cb and Cr components unchanged. After recomposition (Fig. 19, the LQ image no longer displays visible banding, whereas the GT image exhibits clear flicker-banding. These results confirm that the flicker-banding phenomenon originates primarily in the luminance component (Y) and exerts negligible influence on the chrominance components (Cb and Cr).

Building on this luminance-centric insight, we conducted the experiments in Fig. 20. When a synthetic training set was constructed by superimposing the same mask independently on the R, G, and B channels, the resulting models showed no measurable improvement on real-world data. This outcome underscores that simply applying identical per-channel masks in RGB space is not a faithful approximation of the real phenomenon and introduces a substantial domain gap relative to real-world images. Consequently, a fundamental principle for simulating banding in RGB is that one should



Input Reconstructed

Figure 16: The out-of-distribution test results.

avoid superimposing an identical mask across all three channels. Instead, channel-dependent masks must be designed to preserve chrominance while modulating only the luminance (Y).

A.7 Limitations and Areas for Improvement.

Although our proposed method achieves decent performance in flicker-banding removal, there are still some limitations and areas for improvement:

Residual Colored Bands. Some input images may contain extra colored stripes caused by certain display dimming methods. Our model is designed mainly for gray-scale flicker and can leave these colored components visible, as shown in Fig. 21.



Figure 17: Per-channel visualization of banding-free images (GT) and flicker-banding images (LQ) in RGB color space. Flicker-banding artifacts appear in all three channels of the LQ images.



Figure 18: Per-channel visualization of banding-free images (GT) and flicker-banding images (LQ) in YC_bC_r color space. Dark flicker-banding is concentrated mainly in Y, with substantially weaker effects in the chrominance channels.

This limitation arises from the training simulator, which focuses on luminance modulation. Future work will expand the synthetic dataset to cover a wider range of banding types.

Texture Detail Loss. Despite using one-step restoration, the generative diffusion prior can alter original textures or introduce fine details that were not present in the source. As shown in Fig. 22, additional details are generated when removing FB.

This is a common challenge in diffusion-based restoration methods. Future improvements could be made by combining diffusion with other techniques to protect true scene details.

Text-Sensitive Applications. RIFLE can improve visual quality, but character-level fidelity is not guaranteed. The OCR evaluation in Appendix A.11 shows better text fidelity than PiSA-SR but lower fidelity than the original LQ input. Text-sensitive outputs should therefore be verified, and improving character-level fidelity while retaining restoration quality remains an important direction.



Figure 19: Visualization comparison of Y-channel switching effects. *Origin* shows the original image and *Swapped* shows the image after exchanging its Y channel with the counterpart.

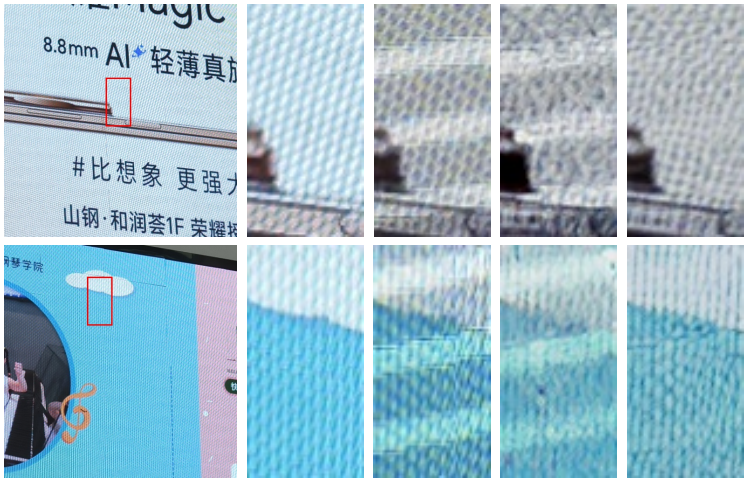


Figure 20: Visual comparison of RIFLE on synthetic datasets based on RGB/YC_bC_r color space.

Handling Severe or Complex Patterns. While our method works well on typical flicker-banding patterns, scenes with highly non-stationary, complex, or curved banding may still pose challenges. Purely black banding regions that completely obscure underlying content can also be difficult to recover. In such cases, the model may failed to restore the lost information.

Improvements could be made by considering multi-frame information, expanding the training data by including more complex banding patterns, or physics-based modeling of display artifacts.

A.8 Our simulated and real-world dataset visualization.

Please refer to Fig. 24-27



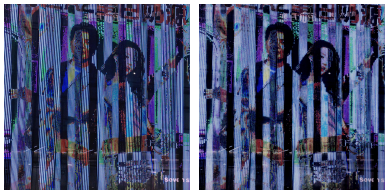
GT LQ RIFLE

Figure 21: A case where our method leaves residual magenta colored bands.



GT LQ RIFLE

Figure 22: RIFLE alters fine texture details.



LQ RIFLE

Figure 23: RIFLE struggles with extremely complex banding patterns.



GT LQ GT LQ

GT LQ GT LQ

Figure 24: Our real-world dataset partial visualization (part 1).



GT LQ GT LQ

GT LQ GT LQ

Figure 25: Our real-world dataset partial visualization (part 2).

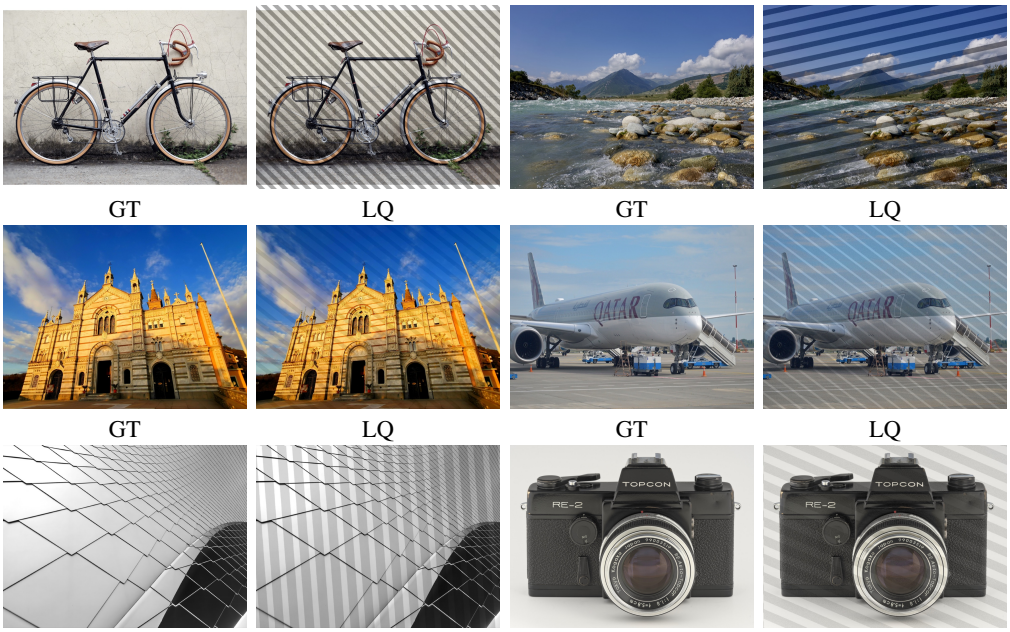


Figure 26: Our simulated dataset based on LSDIR [26] partial visualization (part 1).

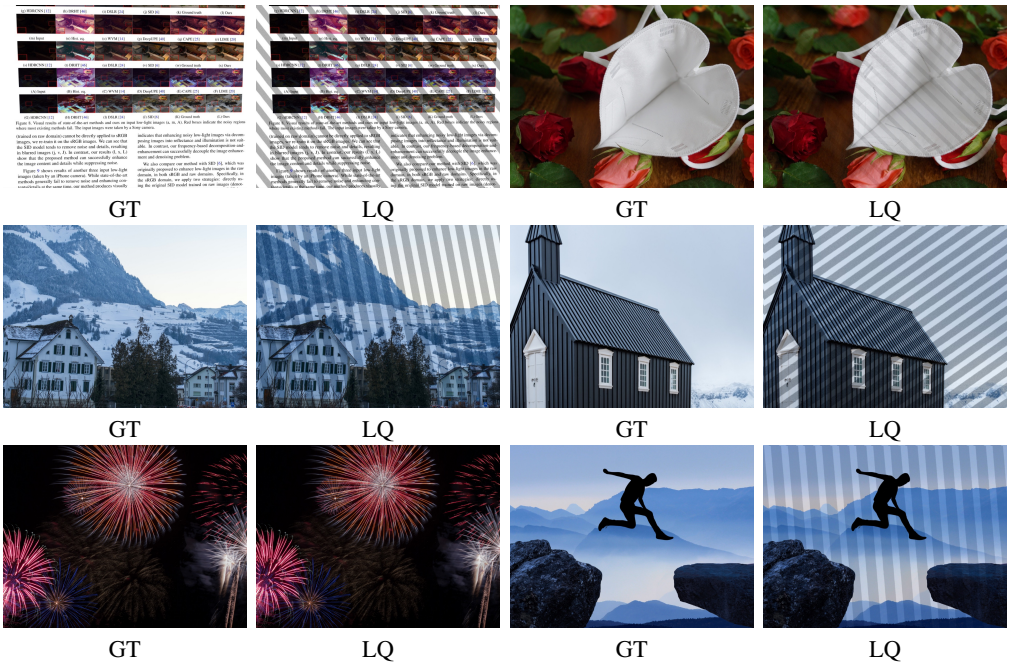


Figure 27: Our simulated dataset based on UHDM [8] partial visualization (part 2).

Table 4: Real-world acquisition settings for the 105 valid image pairs.

Scene	Pairs	LQ exposure	LQ ISO	GT exposure	GT ISO
1	15	1/2500 or 1/500 s	360–1800	1/80 s	250
2	25	1/2500 or 1/800 s	1000–2500	1/60 s	72
3	33	1/2000 s	332–1440	1/80 s	72
4	27	1/2000 s	417–2400	1/80 s	72
5	5	1/1600 s	390–1152	1/80 s	72

Table 5: Training settings and costs for the compared checkpoints. N/A denotes a method without training.

Method	Batch	Iterations	Learning rate	Peak memory	Time (h)
Notch Filter	N/A	N/A	N/A	N/A	N/A
MAT	2	45,000	2×10^{-4}	23.1 GiB	16.2
InvSR	4	4,000	5×10^{-5}	30.8 GiB	20.3
Step1X	8	25,000	10^{-4}	20.5 GiB	69.0
ESDNet	16	186,875	2×10^{-4}	26.4 GiB	50.8
FlickerFormer	1	20,000	5×10^{-5}	17.7 GiB	6.4
PiSA-SR	4	50,000	5×10^{-5}	43.2 GiB	25.7
RIFLE	4	50,000	5×10^{-5}	43.3 GB	25.8

A.9 Real-World Acquisition Settings

The five scenes were captured from commercial LED/OLED-type emissive advertising displays using a vivo X200s under indoor ambient lighting. Table 4 gives the exposure and ISO settings reported for each scene. Short- and long-exposure pairs share a fixed viewpoint, with unchanged ambient lighting. These conditions represent the predefined dark-FB target rather than defining the applicable range after collection. Registration, valid-region cropping, visual screening, and paired patch extraction are described in Sec. 4.

A.10 Training and Inference Costs

Tables 5 and 6 report the training settings used to obtain the compared checkpoints and the measured inference costs. Inference uses the same RTX A6000, 512×512 inputs, and batch size 1. The direct PiSA-SR/RIFLE comparison uses FP16 inference. Memory units in the training table follow the recorded measurements; the RIFLE training value is reported in GB, while the other recorded training values are in GiB.

Relative to PiSA-SR, RIFLE adds 1.239 M parameters, 0.0013 GiB peak memory, 0.002 G profiled MACs, and 0.004 G profiled FLOPs. Latency increases by 38.2 ms (13.7%), mainly because BSHP runs on the CPU; the neural-operation profiles do not capture this CPU preprocessing cost. These measurements characterize patch-level inference under the stated hardware and precision, rather than end-to-end full-resolution mobile inference. RIFLE remains substantially larger and more expensive than lightweight baselines; model compression and efficient prior computation are directions for reducing deployment cost.

Table 6: Inference costs on 512×512 inputs. M and G denote 10^6 and 10^9. Notch filtering runs on the CPU and is not included in GPU-memory or neural-operation profiling.

Method	Params (M)	Memory (GiB)	MACs (G)	FLOPs (G)	Latency (ms)
Notch Filter	0	N/A	N/A	N/A	58.4
MAT	0.690	0.4839	169	337	238.5
InvSR	983.402	3.9520	7,521	15,042	2,607.6
Step1X	12,511.620	39.9809	999,122	1,998,244	24,548.4
ESDNet	5.934	0.2238	70	140	64.1
FlickerFormer	4.155	3.7336	131	263	169.5
PiSA-SR	949.565	4.6580	2,177.477	4,354.953	278.1
RIFLE	950.804	4.6593	2,177.479	4,354.957	316.3

Table 7: OCR-based text fidelity. Higher OCR-A is better.

Metric	LQ	PiSA-SR	RIFLE
OCR-A ↑	0.9594	0.8718	0.8841

Table 8: Full-dataset video evaluation. E_{warp} values are in units of 10^{-3}.

Method	SSIM ↑	MS-SSIM ↑	LPIPS ↓	DISTS ↓	FSIM ↑	GMSD ↓	E_{warp} ↓
LQ	0.6266	0.7018	0.3279	0.1994	0.7668	0.2028	17.6871
PiSA-SR	0.4921	0.6319	0.4077	0.2373	0.7086	0.2418	33.7324
RIFLE	0.6227	0.7074	0.3397	0.2031	0.7709	0.1965	20.5202

A.11 OCR-Based Content Fidelity

Following the OCR-A evaluation protocol used in TADiSR [32], we detect and recognize text in GT and use the recognized strings as reference transcriptions. We reuse exactly the same text regions in LQ, PiSA-SR, and RIFLE and recognize them using the same OCR model. OCR-A converts the Levenshtein edit distance, including insertions, deletions, and substitutions, to a normalized similarity score and averages over the text regions.

RIFLE scores above PiSA-SR but below LQ. Both diffusion restorers therefore incur some text-fidelity loss relative to the input, despite improvements in visual quality. Character-level fidelity is not guaranteed, and outputs intended for text-sensitive use should be verified.

A.12 Video Evaluation and Temporal Consistency

We evaluate PiSA-SR and RIFLE on the complete collected video debanding dataset: 108 videos, 9,849 paired frames, and 9,741 adjacent-frame transitions. All frames use the same 512×512 preprocessing, and both methods restore each frame independently without neighboring frames. We report the six frame-level image-quality metrics and the optical-flow-based warping error E_{warp} for temporal consistency. For each metric, we first average within each video and then report the unweighted mean over the 108 videos.

RIFLE improves all six frame-level metrics relative to PiSA-SR and reduces E_{warp} by 39.2%. Its warping error is closer to LQ, although a gap remains (20.5202 versus 17.6871). These results indicate encouraging temporal stability without an explicit temporal module, rather than guaranteed temporal consistency. Recent video diffusion models use cross-frame aggregation and video-level guidance for space-time super-resolution [33], or spatial and temporal priors for face restoration [34]. A video extension of RIFLE could explore such temporal conditioning while retaining the current frame-restoration objective.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: In the abstract and introduction, we state the main contributions of this paper, including the simulation pipeline, the paired real-world benchmark, and the RIFLE model. We support these claims with the method in Sec. 4 and with the quantitative results, visual comparisons, and ablation study in Sec. 5.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: We discuss limitations and future directions in Appendix A.7. We also note the diversity of display mechanisms and the difficulty of handling more complex banding patterns in practice in Appendix A.1.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

   Justification: Our work is an empirical image restoration study and does not contain theorem-proof style theoretical results.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: We specify the problem setup in Sec. 3, the degradation simulation process, dataset construction, model components, and loss design in Sec. 4, the evaluation setting and ablations in Sec. 5, and additional details in Sec. A. These details are sufficient to understand and reproduce our main experiments.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No].

   Justification: We commit to publicly releasing the dataset and model code at `https://github.com/libozhu03/RIFLE`, as stated in the abstract; a release plan alone does not establish current open access.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: We describe training data construction, model design, and loss design in Sec. 4; test data, evaluation metrics, implementation details, and ablation settings in Sec. 5; and supplementary details in Sec. A. We also report the key hyperparameters needed to interpret the results.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: Appendix A.4 reports a decision-level 95% Wilson confidence interval and a one-sided exact binomial test for user preferences, with the analysis unit and its limitations stated explicitly. Restoration metrics are single-run results; we do not claim across-run confidence intervals for them.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No].

   Justification: Appendix A.10 reports training settings, peak memory, training time, and inference profiling on an RTX A6000 with the input size and batch size specified. However, the complete training hardware configuration and total development compute are not documented, so we retain the conservative [No] answer.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes].

Justification: To the best of our knowledge, our research complies with the NeurIPS Code of Ethics. We study an image restoration problem, credit prior assets, and do not involve sensitive personal data collection or high-risk deployment claims.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No].

    Justification: Our current manuscript does not include a dedicated discussion of broader societal impacts. Although this work can improve screen-photography readability and usability, we do not yet explicitly discuss potential positive and negative downstream impacts.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: We do not release a high-risk generative model, scraped dataset, or other asset that would typically require special misuse safeguards. Our contribution is an image restoration method and a paired benchmark for flicker-banding removal.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No].

    Justification: We cite the original datasets and baseline methods that we build on, but we do not yet explicitly document the licenses or terms of use of all existing assets in the manuscript or appendix.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No].

    Justification: We introduce a new paired real-world dataset and a code release plan, but documentation for these assets is not yet provided alongside an anonymized release in the current submission materials.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [No].

    Justification: We include a user study and provide screenshots of the evaluation interface in Appendix A.4, but we do not include the full participant instructions or compensation details in the current version.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No].

Justification: We include a user study, but our current manuscript does not describe participant-risk disclosure or any IRB or equivalent review status.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: LLMs are not an important or non-standard component of our proposed method, experiments, or scientific claims.