RGB-EVENT ISP: THE DATASET AND BENCHMARK

Anonymous authors
Paper under double-blind review

ABSTRACT

Event-guided imaging has received significant attention due to its potential to revolutionize instant imaging systems. However, the prior methods primarily focus on enhancing RGB images in a post-processing manner, neglecting the challenges of image signal processor (ISP) dealing with event sensor and the benefits events provide for reforming the ISP process. To achieve this, we conduct the first research on event-guided ISP. First, we present a new event-RAW paired dataset, collected with a novel but still confidential sensor that records **pixel-level aligned** events and RAW images. This dataset includes 3373 RAW images with 2248×3264 resolution and their corresponding events, spanning 24 scenes with 3 exposure modes and 3 lenses. Second, we propose a conventional ISP pipeline to generate good RGB frames as reference. This conventional ISP pipleline performs basic ISP operations, *e.g.*demosaicing, white balancing, denoising and color space transforming, with a ColorChecker as reference. Third, we classify the existing learnable ISP methods into 3 classes, and select multiple methods to train and evaluate on our new dataset. Lastly, since there is no prior work for reference, we propose a simple event-guided ISP method and test it on our dataset. We further put forward key technical challenges and future directions in RGB-Event ISP. In summary, to the best of our knowledge, this is the very first research focusing on event-guided ISP, and we hope it will inspire the community.

1 INTRODUCTION

Since their invention in 1975, digital cameras have profoundly impacted various aspects of modern society (Delbracio et al., 2021; Kyung et al., 2016). Active pixel sensors (APS) (Liebe et al., 1998) are used as the core of cameras to capture RGB color signals, recording images or videos. This technology forms the foundation for widespread applications in smartphones (Delbracio et al., 2021), autopilot systems (Ingle & Phute, 2016), drones (Zhu et al., 2018), virtual reality (Huang et al., 2017), and more. However, nowadays APS has reached a bottleneck *wrt.* power consumption, frame rate, and dynamic range due to its global recording characteristics (Gallego et al., 2020). Event vision sensors (EVS), with their inherent asynchronous recording property, achieve lower power consumption ($< 10mW$), lower latency ($< 1ms$), and higher dynamic range ($> 120dB$) (Gallego et al., 2020). As a result, integrating EVS as a significant enhancement to APS imaging system has received considerable attention in recent years (Lu et al., 2023b; Tulyakov et al., 2021; Gallego et al., 2020; Tulyakov et al., 2022). Heavy efforts have been put on developing new imaging system combining EVS and APS (Shariff et al., 2024; Lu et al., 2023b;a). The introduction of EVS has nearly reshaped the entire framework of imaging formation and enhancement, impacting almost all relevant areas *e.g.*, video super-resolution (Lu et al., 2023b; Jing et al., 2021), video frame interpolation (Tulyakov et al., 2021; 2022; Lu et al., 2023a), deblurring (Yuan et al., 2007; Zhang et al., 2022; Yunfan et al., 2023), high dynamic range imaging (Xiaopeng et al., 2024; Messikommer et al., 2022), low-light image enhancement (Wang et al., 2020b; Liang et al., 2024), and rolling shutter correction (Zhou et al., 2022; Lu et al., 2023a). *However, the majority of previous work focuses on using events as auxiliary information to boost the performance of classical RGB imaging systems, while methods and benchmarks that considering the challenges and opportunities of events in the APS ISP process, are lacking.*

Merging APS and EVS in ISP is non-trivial on the implementation level. Prism spectrometer is an early stage attempt and it needs the corresponding optical mechanic setting (Tulyakov et al., 2022). However, this prism-based approach is very cumbersome, requiring additional optical prisms and

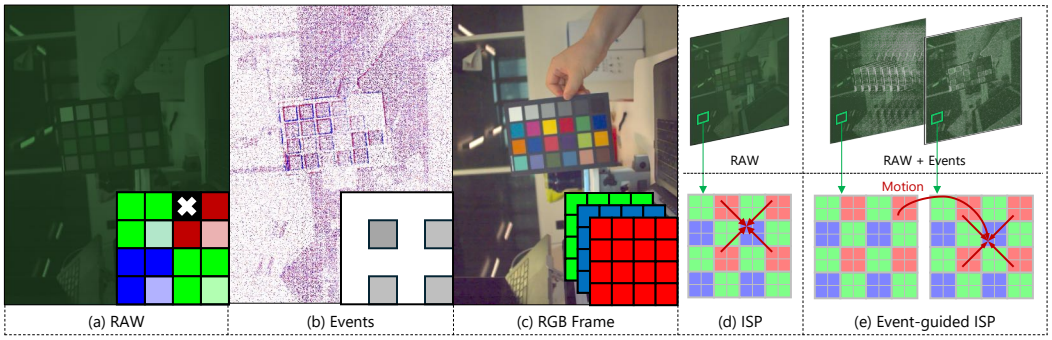


Figure 1: (a), (b), and (c) display a RAW, Events, and RGB frame captured by the hybrid vision sensor (HVS), respectively. The RAW image follows a quad-Bayer pattern (Yang et al., 2022), while the events are positioned at the lower-right corner of each color pixel block, making the RAW resolution twice that of the events. (d) illustrates the traditional ISP process. (e) shows the potential event-guided ISP process, where the higher temporal resolution of events can captures motion information for ISP.

failing to ensure the alignment between APS and EVS. Sensors that integrate both APS and EVS on the photodiode level are referred to as hybrid-vision sensors (HVS) (Yaqi et al., 2024; MIPI Challenge 2024, 2024), which represent a cutting-edge technology, offering significant advancements in camera imaging. Due to the manufacturing complexity and error-prone design process of HVS, the RAW data generated by APS in HVS exhibits higher noise, missing values at fixed positions, and is more sensitive to defects (MIPI Challenge 2024, 2024; Yaqi et al., 2024). Recent works have acknowledged this challenge and proposed datasets for demosaicing, denoising, or defect correction for APS RAW, where *the challenges in APS of HVS take precedence over the potential benefits events signal could provide.* With the inherent higher dynamic range and lower latency, events can perceive a broader spectrum and capture more-instant motion information (Shekhar Tripathi et al., 2022; Liang et al., 2021), allowing significant potential for boosting the denoising and color correction of ISP processing of APS RAW, as shown in Fig. 1.

To better explore the benefits of events on the ISP process of HVS, we propose a new dataset with **pixel-wise aligned** events and APS RAW image. This dataset uses the under-development HVS-ALPIX-Eiger sensor (Alpsentek, 2024), which rearranges event and APS in a quad-Bayer pattern (a quarter photodiodes are dedicated for event, as in Fig.1). This sensor has a high resolution with 1224×1632 for events and 2248×3264 for RAW, and offers superior color and noise profiles compared to the DVS346 (Scheerlinck et al., 2019). These features make it promising for various applications (Lu et al., 2023b).

We ensure the dataset diversity in two ways: photographic setting and scenes. For photographic setting, we adopt various values of aperture, focal length and exposure time. For the scene diversity, we cover 12 categories of scenes, across a wide range of color scenes, including flowers, buildings, under different weather and lighting conditions. In total, 3373 APS frames and the corresponding events are captured. A standard 24-color ColorChecker (Goto et al., 2003) is applied at certain frames as the color correction reference, as shown in Fig. 1 (c).

To generate the ground truth RGB images for the dataset, we propose a controllable ISP framework based on MATLAB (Poon & Banerjee, 2001). This ISP framework, using the ColorChecker as a prior, performs tasks such as black level calculation (Li et al., 2010), demosaicing (Hirakawa & Parks, 2006), white balance (Weng et al., 2005), denoising (Abdelhamed et al., 2018), and color correction (McElvain & Gish, 2013), resulting in high-quality RGB images with controllable errors as the reference ground truth. Since the controllable framework requires the ColorChecker information as a prior, it cannot generalize to arbitrary scenes. The color accuracy and temporal stability of this ISP are also analyzed.

We categorize the existing ISP methods with RAW input into three categories and benchmark their performances on our dataset. We compare their performances across various scenarios and further conduct analysis on certain phenomena we have observed. Additionally, we propose a simple UNet-like (Ronneberger et al., 2015) event-guided ISP neural network to fuse events with RAW images.

This simple network can effectively improve the outdoor performance of ISP compared to the original UNet (Ronneberger et al., 2015). We also identify key contributions and challenges of events in the ISP process, providing a foundation and direction for future research.

2 Related Works

Event-guided Imaging Datasets: Event camera-guided imaging enhancement is an emerging field where the contribution of real datasets is crucial (Gallego et al., 2020). Currently, event cameras have made significant progress in areas such as frame interpolation (Tulyakov et al., 2021; Lu et al., 2023a; Niklaus et al., 2017; Bao et al., 2019), video super-resolution (Lu et al., 2023b; Jing et al., 2021), low-light enhancement (Liang et al., 2024; 2023), and deblurring (Xu et al., 2021; Lin et al., 2020; Jiang et al., 2020). These advancements are supported by many foundational datasets (Tulyakov et al., 2021; Scheerlinck et al., 2019; Lu et al., 2023b). For example, BS-REGB (Tulyakov et al., 2022) is a frame interpolation dataset using a beamsplitter to pair event cameras and RGB cameras. The CED (Scheerlinck et al., 2019) dataset and APLEX-VSR (Lu et al., 2023b) dataset have been used in research on event camera-guided video super-resolution. Overall, these datasets serve as the cornerstone and pioneers in research on related tasks. *However, these datasets assume that event cameras can obtain high-quality RGB images through the ISP process, an assumption that is often too idealistic.* Recognizing this, the MIPI (Yaqi et al., 2024; MIPI Challenge 2024, 2024) challenge introduced a RAW demosaic dataset for HVS in event cameras, addressing challenges like high noise and missing values in RAW from HVS. *Although this dataset is the first to focus on the RAW domain ISP process in event cameras, it lacks real event streams, thereby overlooking the potential role of events in the ISP process*. To address this gap, we propose the ***first*** dataset with aligned RAW and events from a new HVS, aiming at exploring the potential value and role of event data in the ISP process.

Learning-based ISP: Traditional ISPs (Schwartz et al., 2018) consist of long pipelines. In recent years deep learning has brought new insights to ISPs (da Silva et al., 2023a) and has achieved higher performance. These methods can be roughly categorized into three types. The first type is full pipeline replacement methods, such as PyNet (Ignatov et al., 2020b) which use CNN architectures to replace the entire ISP pipeline. The second type is stage-wise enhancement methods, like CameraNet (Liang et al., 2021) and AWNet (Dai et al., 2020), which divide the ISP pipeline into restoration and enhancement stages. The third type is image enhancement network-based methods, which utilze state-of-the-art image proessing backbone models such as UNet (Ronneberger et al., 2015) and Swin-Transformer (Liu et al., 2021) to deal with ISP tasks. Though these methods have proven effective for RAW to RGB conversion, the potential of events in this process is not explored.

Event-guided Image/Video Enhancement: Due to their high dynamic range and high temporal resolution (Gallego et al., 2020; Shariff et al., 2024), event cameras have garnered significant attention in the field of image/video enhancement and restoration (Gallego et al., 2020; Shariff et al., 2024), including many applications. Initially, the use of events focused primarily on single-task enhancements of RGB images or videos (Tulyakov et al., 2021; Pan et al., 2019; Lu et al., 2023b). Recently, researchers recognized image enhancement tasks are inherently coupled with various degradations interwoven (Zhang & Yu, 2022; Song et al., 2022; Yunfan et al., 2023), suggesting a trend towards using events for unified solutions in camera computational imaging for multiple tasks. *However, existing methods focus* ***solely*** *on enhancing RGB images or videos using events, overlooking the ISP pipeline, which generate RGB images from RAW images. Additionally, existing methods neglect the potential value that events could provide in the ISP process.*

3 Dataset Collection

As the first dataset, which we call HVS-ISP Dataset, featuring paired raw-event data collected using a HVS, our aim is to facilitate research on event-guided RAW ISP. We selected the HVS-Eiger sensor developed by ALPIX (Alpsentek, 2024), which can output both APS and EVS signals that align in both time and space, as show in Fig. 2 (b). More parameter details of APS and EVS are shown in Tab. 1. Compared to the Prophesee sensor (Tulyakov et al., 2021), which can only output event signals, and the DVS346 sensor (Scheerlinck et al., 2019), which has lower resolution (260×346) and higher noise, our choice offers significant advantages. Hence our dataset, captured with this

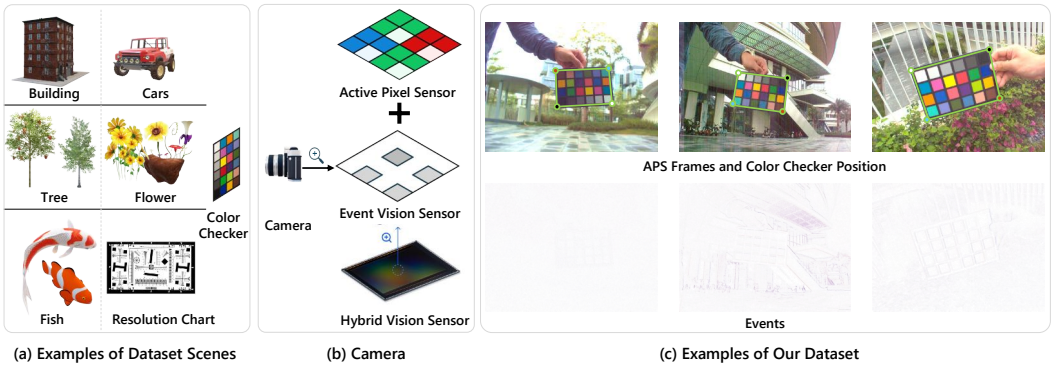


Figure 2: Overview of dataset collection. (a) illustrates the variety of scenes in the dataset, including buildings, plants, animals, and calibration boards. (b) presents a schematic of the HVS sensor, composed of a stacked active pixel sensor (APS) and an event vision sensor (EVS). (c) displays dataset samples.

Table 1: Comparison between active pixel sensor (APS) and event vision sensor (EVS) (Alpsentek, 2024) in our dataset collection. APS and EVS are stacked together to form a hybrid-vision sensor (HVS).

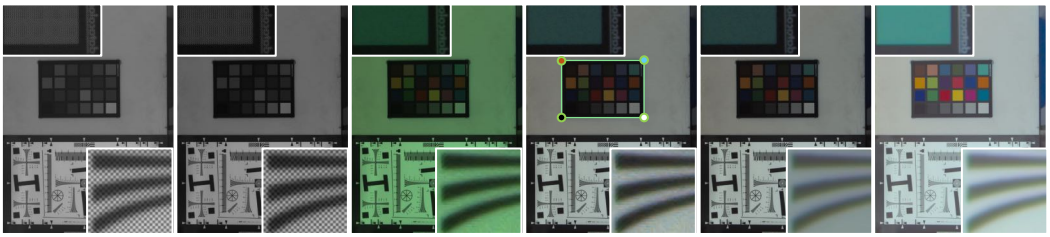
Sensor	Resolution	Frame Rate	Power Consumption	Redundant Data Rate	Dynamic Range
APS	2248×3264	10∼60 fps	$>$ 100 mW	10 MB/s	60 dB
EVS	1124×1632	$\geq$ 800 fps	∼10 mW	40-180 KB/s	$>$ 120 dB

advanced new sensor, holds significant value for the event vision research, providing a foundation resource for advanced exploration in event-guided RAW ISP.

The collection of the dataset focuses on two main aspects: **(1)** the **diversity** of the dataset, ensuring it has broad representativeness to cover a wide range of real-world scenarios; **(2)** the inclusion of a **ColorChecker** for ISP calibration, which helps the ISP accurately restore scene colors to generate high-quality RGB frames as references.

(1) Dataset Diversity: In constructing our dataset, we paid particular attention to two types of diversity: camera parameter diversity and scene diversity. ***Camera Parameter Diversity:*** To ensure that our dataset encompasses a variety of photographic conditions, we made extensive adjustments to the camera parameters. This included aperture values ranging from $F1.0$ to $F6.0$, focal lengths extending from $8mm$ to $52mm$, and exposure times varying from $1ms$ to $100ms$. ***Scene Diversity:*** We focused on three key aspects to ensure comprehensive scene diversity: *Light Source Diversity:* We distinguished between indoor artificial light and outdoor natural light, with special consideration for different weather conditions. Data collection was performed under various lighting conditions, including sunny and cloudy days. *Motion Diversity:* We captured both dynamic and static videos, ensuring a mix of scenes with and without motion blur. This variety helps in testing and enhancing the performance of image processing algorithms under different motion conditions. *Material Diversity:* We included a wide array of scenes such as trees, plants, buildings, fish, dolls, and more. These scenes exhibit a broad spectrum of colors and textures, providing a rich dataset for comprehensive testing and improvement of image processing techniques.

(2) ColorChecker as ISP Reference: To ensure precise color correction and white balance in ISP pipeline, we utilized a standard 24-color ColorChecker (Tian et al., 2002) as critical references. At the start of each video shoot, we captured frames containing the ColorChecker and gradually removed the chart from subsequent frames. We meticulously annotated the position of the ColorChecker in each frame using the LabelMe tool (Russell et al., 2008), as shown in Fig. 2 (c). For frames without the ColorChecker, we applied previously determined ColorChecker parameters as references. This approach guarantees reliable color correction data in our dataset. Incorporating the ColorChecker allows generating high-quality RGB values, enhancing color fidelity. This method



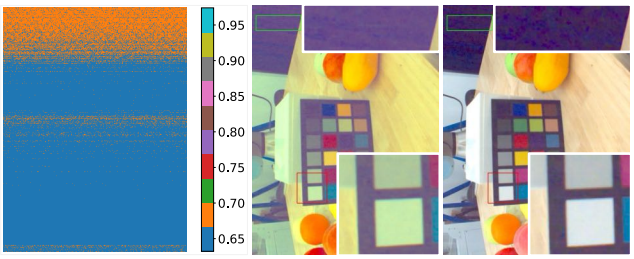
(a) Quad-bayer pattern raw (b) Black pattern, Fixed-pattern (c) Demosacing (d) White Balancing (e) Denoise (f) Color space transform & Gamma

Figure 3: Flows in controllable ISP process. (a) Quad-bayer pattern raw image, which serves as the initial input. (b) Black pattern and fixed-pattern noise removal to suppress sensor-induced artifacts. (c) Demosaicing to reconstruct a rgb image from the raw data. (d) White balancing using a ColorChecker for accurate color reproduction. (e) Denoising to filter out spatial noise from the image. (f) Color space transformation and Gamma to convert the image into the desired color space for final output.

ensures robustness for applications requiring accurate color restoration. Additionally, we conducted a thorough manual review of the ColorChecker annotations to validate their accuracy, further improving our dataset's reliability for ISP algorithms.

In summary, based on these two main objectives, we captured a total of 24 videos. Each video contains 80 to 140 frames, resulting in a total of 3373 APS RAW frames and their corresponding events. Additionally, the dataset includes the positions of the ColorCheckers within the APS images. We divided the dataset into training and test sets, with $3/4$ of the data used for training and $1/4$ for testing. The testing set includes 3 indoor scenes and 3 outdoor scenes to ensure sufficient diversity. *For more details on data collection and visualizations, please refer to the supplementary material.*

4 Controllable ISP



(a) Fixed Pattern Noise (FPN) (b) ISP w/o remove FPN (c) ISP w remove FPN

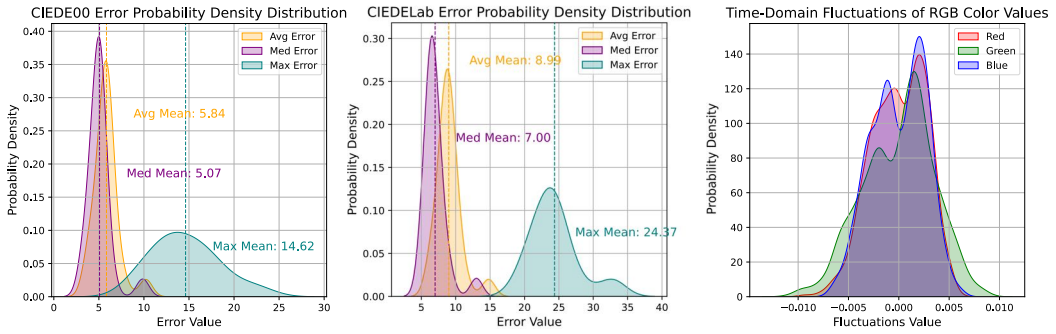
Figure 4: Fixed pattern noise (FPN) removal. (a) Visualizes the camera's fixed pattern noise. (b) and (c) show the RGB images without and with fixed pattern noise removal, respectively. The image in (c) demonstrates lower noise and more accurate white balance after the removal of fixed pattern noise.

The controllable ISP aims to provide module-based and analytically measurable RGB frames based on the APS RAW. With the support of the contained ColorChecker, the resulting frames have good color accuracy and low noise, serving as the reference for APS. Requirement of the ColorChecker prevents from generalizing to other arbitrary scenes. In this section, we introduce each module, followed by a quality evaluation and pros-and-cons discussion, with the hope that this ISP pipeline will be beneficial for the community.

4.1 Controllable ISP Pipeline

Fig. 3 depicts that how an image is processed via a conventional ISP pipeline, making the reference for the APS data. **(1) Black Level and Fixed Pattern Subtraction:** Taking an arbitrary unprocessed bayer raw as input, a pre-calibrated global black level value *blc* is subtracted, following by subtracting a fixed pattern vector *fpn* [1]. *blc* is the min of a raw image taken under a pure-black environment while *fpn* is a vector that records the per-row average value as the used sensor is only with horizontal fixed pattern, as shown in Fig. 4.

[1] *blc* and *fpn* are calibrated in a pure-dark laboratory setting. Over five frames are captured and averaged to increase the calibration accuracy.



(a) CIEDE00 Error (b) CIELAB Δ_{ab} Error (c) Time Domain Fluctuations

Figure 5: Color errors and fluctuations of our ISP method, computed using a ColorChecker. **(a)** CIEDE 2000 Error Probability Density Distribution: Displays CIEDE 2000 error values distribution with annotations for average (5.84), median (5.07), and maximum error means (14.62). **(b)** CIEDE Lab Error Probability Density Distribution: Shows CIEDE Lab error values distribution, indicating average (8.99), median (7.0), and maximum error means (24.37). **(c)** Time-Domain Fluctuations of RGB Color Values: Illustrates RGB color values fluctuations over time, representing temporal stability and variations in color accuracy.

(2) Demosaicing: Given bayer pattern, the well-adopted demosaicing method (Rainbow-Johnny-Johnny-Image-Processing-Lim, 2022) is used. The resolution is preserved while the channel number is tripled. Note that this method is still prone to generating false color in very high frequency area, as shown in Fig. 3(c). **(3) Manual White Balancing:** On a RGB image (greenish due to no white balance), we use LabelMe (Russell et al., 2008) to extract the mean colors of 24 ColorChecker patches. The 21_{st} patch is used as the groundtruth illumination for manual white balance. **(4) Spatial Denoising:** We use a milestone denoising method BM3D (Dabov et al., 2009) to perform spatial denoising with the setting of $\sigma = 50$. **(5) Color Space transform:** Following Finlayson et.al. (Finlayson et al., 2015), given the retrieved ColorChecker values and the predefined oracle ColorChecker values, we optimize towards the CIEDE00 error and obtain the final color correction matrix *ccm* of the shape $(3, 3)$. A linear sRGB image is then computed from the input image I: $I_{linsrgb} = I * ccm$. **(6) Gamma:** Following sRGB standard (Anderson et al., 1996), a piecewise gamma curve is applied for brightness perception. *Due to space limitations, please refer to the supplementary material for more details and hyperparameters of controllable ISP.*

4.2 Controllable ISP Evaluation

We evaluated the controllable ISP in two main aspects: the **color accuracy** of individual images and the **temporal stability** of color recovery in continuous videos. For color accuracy, we used the CIEDE00 (Luo et al., 2001) and CIELAB Δ_{ab} (Lee & Powers, 2005) metrics to evaluate color accuracy. CIEDE00 is a widely used metric for color matching, considering the nonlinear characteristics of color differences and the human eye's sensitivity to colors, which accurately reflects human visual perception of color differences. CIELAB Δ_{ab} is a color difference metric based on the CIELAB color space (Mahy et al., 1994). Specifically, as shown in Fig. 5 (a) (b), we conducted a ColorChecker-based evaluation on 100 randomly selected samples. In CIEDE00 (Luo et al., 2001), we obtained an average value of 5.84 and a median value of 5.07; For CIELAB Δ_{ab}, we obtained an average value of 8.99 and a median value of 7.00, demonstrating that our method can generally restore colors up to an accurate level. We displayed the maximum error distribution per image, showing that in CIEDE00 it is around 14, and in CIELAB Δ_{ab} around 24, affected by color filter sensitivity and photodiode layout. For temporal stability in frame estimation differences, as shown in Fig. 5 (c). We selected a 140-frame video, marking the ColorChecker in each frame. After generating colors frame by frame, we observed that differences for the 24 ColorChecker colors are all under 0.01, mostly within 0.005. This confirms our algorithm's temporal stability.

In summary, we presented a controllable ISP pipeline and analyzed its performance. However, the ISP contains numerous controllable variables and hyperparameters. We hope that future researchers will focus on optimizing these controllable aspects of the ISP to further enhance its performance.

Table 2: Comparison on Parameters, FLOPS, and Time. Top two models are highlighted in **red** and **green**.

	Unet	PyNet	CameraNet	AWNet	PyNetCA	MW-ISPNet	InvertISP	Swin Transformer	eSL	Ev-UNet
Params↓	16.64	47.55	25.79	96.07	29.27	**7.22**	92.44	8.87	**0.737**	21.51
GFLOPS↓	**4.52**	111.96	19.19	120.21	51.27	29.22	**1.41**	14.24	48.49	6.89
Time (s)↓	**0.0100**	0.0775	0.0300	0.2138	0.0308	0.0459	0.0436	0.0868	0.063	**0.012**

Table 3: Comparison of Methods on HVS ISP Dataset outdoor scenes. Top two models are highlighted in **red** and **green**. * refer to the results obtained by the same model with different hyperparameters.

	2-Out-Tree-2			3-Out-Flower-2			4-Out-Building-1			Average		
	PSNR↑	**SSIM**↑	L_1↓	**PSNR**↑	**SSIM**↑	L_1↓	**PSNR**↑	**SSIM**↑	L_1↓	**PSNR**↑	**SSIM**↑	L_1↓
PyNET	**31.70**	**0.9818**	**0.0190**	**35.12**	**0.9784**	**0.0127**	**30.60**	**0.9752**	**0.0223**	**32.47**	**0.9785**	**0.0180**
PyNET*	27.56	0.9711	0.0310	32.35	0.9646	0.0175	28.20	0.9600	0.0311	29.37	0.9652	0.0265
PyNetCA	**31.86**	**0.9788**	**0.0202**	**34.19**	**0.9773**	0.0139	**29.22**	**0.9725**	0.0280	**31.76**	**0.9762**	**0.0207**
InvertISP	28.56	0.9487	0.0243	25.59	0.9298	0.0313	28.62	0.9307	0.0287	27.59	0.9364	0.0281
MV-ISPNet	27.05	0.9680	0.0256	33.61	0.9648	**0.0137**	28.62	0.9657	0.0304	29.76	0.9662	0.0232
CameraNet	11.18	0.2580	0.2289	12.39	0.2741	0.1899	10.52	0.2534	0.2609	11.36	0.2618	0.2266
CameraNet*	13.26	0.637	0.2044	13.59	0.2736	0.1770	10.06	0.2753	0.2474	12.30	0.3953	0.2096
AWNet	14.33	0.8836	0.1166	20.10	0.9316	0.0519	16.70	0.9390	0.0951	17.04	0.9180	0.0879
Swin-Transformer	25.02	0.9539	0.0308	29.14	0.9555	0.0231	21.57	0.9295	0.0523	25.24	0.9463	0.0354
UNet	21.97	0.9583	0.0393	29.43	0.9717	0.0208	22.12	0.9603	0.0460	24.51	0.9634	0.0354
UNet*	29.52	0.9752	0.0206	25.75	0.9623	0.0323	29.24	0.9680	**0.0265**	28.17	0.9685	0.0265
eSL-Net	25.67	0.9424	0.0342	19.39	0.9180	0.0576	24.01	0.9277	0.0502	23.02	0.9294	0.0473
EV-UNet	32.86	0.9795	0.0148	32.87	0.9698	0.0157	24.59	0.9600	0.0369	30.11	0.9698	0.0225

5 Benchmark and Direction

Based on the RGB frames obtained from the controllable ISP, we evaluate the performance of four types of ISP methods, particularly in outdoor and indoor scenarios. The experiments are conducted in the same environment and framework. Additionally, we will discuss the potential reasons behind these results and propose future research directions.

Implementation Details: All our models were trained and tested on the same machine with a single A40 GPU with 48GB of GPU memory. We used PyTorch (Paszke et al., 2017) for all experiments, applying random cropping and rotation for data augmentation. The training batch size was 1, with each patch sized at 1024×1024. The learning rate was 0.0001, and all models were trained for 50 epochs.

Evaluation Metrics: We evaluate model performances in two aspects: resource consumption, including parameters in millions (M), GFLOPS, and average inference time (s); and image reconstruction for indoor and outdoor scenes, measured by PSNR (Hore & Ziou, 2010), SSIM (Brunet et al., 2011), and L_1 distance.

5.1 ISP Benchmark Methods

Inspired by the prior ISP survey study (da Silva et al., 2023b), we categorize learning-based ISP models into three classes: full pipeline, stage-wise, image enhancement network-based. We selected two to four open-source models from each category for training and evaluation on our dataset. Furthermore, we put forward another new category of event fusion method, and since there is no prior research to refer to, we design a simple event-guided ISP neural network to test on our dataset. *For more details on ISP methods, please refer to the supplementary material.*

Full Pipeline ISP: These models utilize CNN architectures to integrate traditional ISP processes into an end-to-end conversion from RAW to RGB images. Notable models in this category include PyNet (Ignatov et al., 2020b), PyNetCA (Kim et al., 2020), InvertISP (Xing et al., 2021), and MV-ISPNet (Ignatov et al., 2020a).

Stage-wise ISP: They employ multiple specialized modules to handle different ISP tasks, either sequentially or in parallel, to produce the final image. In our benchmark, we selected CameraNet (Liang et al., 2021) and AWNet (Dai et al., 2020) for their distinct approaches. Note that due to the unavailability of a PyTorch version of CameraNet (Liang et al., 2021), we experimented on a

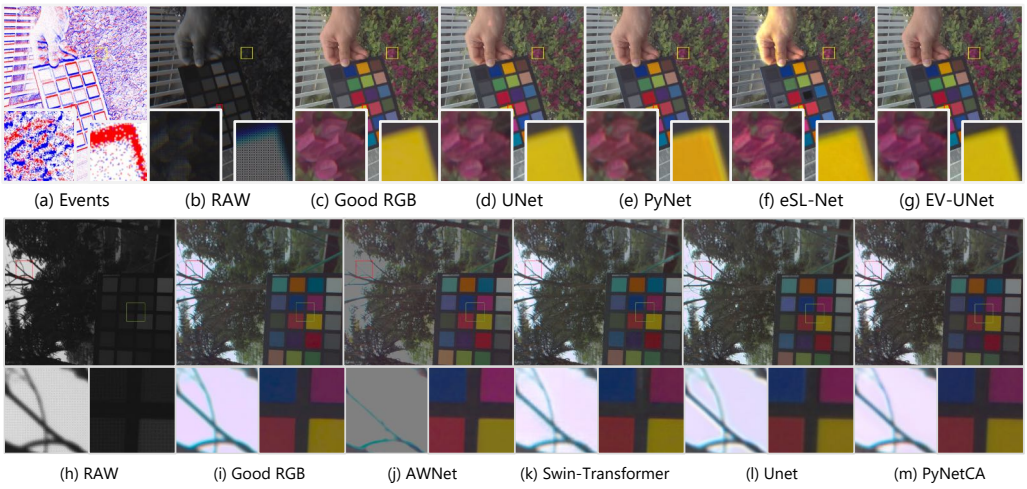


Figure 6: Visualization results of different methods on HVS-ISP Dataset outdoor scenes.

Table 4: Comparison of Methods on HVS ISP Dataset indoor scenes. * refer to the results obtained by the same model with different hyperparameters.

	1-In-Fruit-2			3-In-ColChecker-40			4-In-RLChart-10			Average		
Methods	**PSNR↑**	**SSIM↑**	L_1↓	**PSNR↑**	**SSIM↑**	L_1↓	**PSNR↑**	**SSIM↑**	L_1↓	**PSNR↑**	**SSIM↑**	L_1↓
PyNET	13.09	0.7970	0.2182	11.38	0.7922	0.2489	11.42	0.7100	0.2563	11.97	0.7664	0.2412
PyNET*	14.46	0.8068	0.2008	24.02	0.9550	0.0497	13.58	0.7694	0.1978	17.36	0.8437	0.1494
PyNetCA	18.13	0.8843	0.1253	29.53	0.9723	0.0246	**35.51**	**0.9727**	**0.0121**	27.72	0.9431	0.0540
InvertISP	25.83	0.9098	0.0346	28.33	0.9500	**0.0235**	30.65	0.9578	0.0183	28.27	0.9392	0.0254
MV-ISPNet	**31.91**	**0.9594**	**0.0185**	**29.56**	**0.9729**	0.0265	31.88	0.9670	0.0170	**31.12**	**0.9664**	**0.0207**
CameraNet	13.06	0.2660	0.1947	13.58	0.2722	0.1836	12.47	0.2391	0.2257	13.04	0.2591	0.2013
CameraNet*	14.18	0.290	0.1630	10.60	0.2667	0.2545	13.26	0.2636	0.2044	12.68	0.2672	0.2073
AWNet	17.95	0.8665	0.1302	**32.17**	**0.9807**	**0.0184**	30.98	0.9596	0.0215	27.03	0.9356	0.0567
Swin-Transformer	25.73	0.9397	0.0301	25.50	0.9561	0.0359	26.18	0.9486	0.0252	25.80	0.9481	0.0304
UNet	17.62	0.9161	0.0747	13.96	0.8828	0.1454	15.53	0.8750	0.1170	15.70	0.8913	0.1124
UNet*	**32.52**	**0.9659**	**0.0161**	29.04	0.9740	0.0257	**33.72**	**0.9716**	**0.0146**	**31.76**	**0.9705**	**0.0188**
eSL	27.09	0.9428	0.0331	24.79	0.9548	0.0434	26.52	0.9415	0.0379	26.13	0.9464	0.0381
EV-UNet	14.16	0.8706	0.1533	31.64	0.9779	0.0214	32.33	0.9678	0.0173	26.04	0.9388	0.0640

converted version. The modules in the original AWNet (Dai et al., 2020) are trained independently, however in our experiment we trained them end-to-end.

Image Enhancement Network-Based ISP: There have been numbers of high performance backbone models for image enhancement in image enhancement tasks like deblurring (Zhang et al., 2022) and super-resolution (Chen et al., 2022). Though not initially designed for ISPs, minor modifications can adapt these models for ISP tasks. For our benchmark, we selected UNet (Ronneberger et al., 2015) and Swin-Transformer (Liu et al., 2021; Lu et al., 2024).

Event Fusion Method: As the first research on event-guided ISP, we have no prior research for reference. Therefore, we selected eSL-Net (Wang et al., 2020a), an event-based backbone network used in various tasks (Lu et al., 2023b). Additionally, we merged events as voxel-grid (Liu et al., 2023) with UNet's encoder as EV-UNet to verify events effectiveness and challenges.

5.2 Comparative Experiments and Visualization Analysis

Computational Performance: In Tab. 2, InvertISP (Xing et al., 2021) excels in computational efficiency with 1.41 GFLOPS, significantly lower than the over 100 GFLOPS of AWNet (Dai et al., 2020) and PyNet (Kim et al., 2020), which is suitable for limited computing resources. UNet surpasses CameraNet (Liang et al., 2021) in processing speed with a response time of 0.01 s, preferable for real-time performance. Overall, UNet demonstrates balanced performance with low GFLOPS and the fastest processing speed, due to its straightforward design.

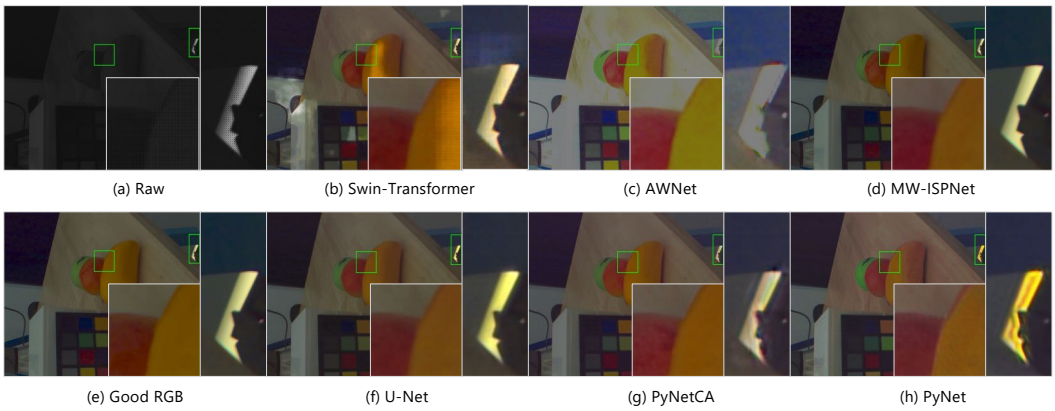


Figure 7: Visualizations on HVS-ISP Dataset indoor scenes.

Outdoor Performance: Tab. 3 shows the superior performance of PyNet across three outdoor backgrounds. PyNet (Kim et al., 2020) achieves the best PSNR (Hore & Ziou, 2010), SSIM (Brunet et al., 2011), and L_1 with an overall average PSNR (Hore & Ziou, 2010) of 32.47, significantly higher than other models. Specifically, EV-UNet shows significant improvement in outdoor scenes with UNet after incorporating events gain, increasing from 28.17 to 30.11. In contrast, the commonly used event-based method eSL-Net performs poorly with a PSNR (Hore & Ziou, 2010) of only 23. This poor performance mainly results from the ***limited receptive field*** of eSL, which is insufficient for estimating the ***global illumination*** information, and thus failing to achieve consistent global illumination enhancement. We further discuss on this issue in Sec. 5.3. we also visualize the results in Fig. 6. PyNet has achieved the highest PSNR (Hore & Ziou, 2010) but exhibits edge artifacts, this is likely due to the overfitting of the model. In outdoor scenes, event-enhanced outputs of EV-UNet show good global consistency. Fig. 6 shows that AWNet (Dai et al., 2020) struggles with fine texture restoration, explaining its inferior performance to other methods.

Indoor Performance: Tab. 4 shows that UNet* excels in indoor environments, especially when handling multiple colored fruits and scenes with complex lighting and details. The output of AWNet (Dai et al., 2020) has overall excessive brightness, as illustrated in Fig. 7, explaining its low PSNR values. PyNet exhibits noticeable artifacts, consistent with the good RGB edge but with significantly different brightness, likely due to the ill-posed nature of brightness recovery in the ISP process, resulting in its poor indoor performance. Event-fusion methods perform poorly indoors, primarily due to flickering light sources that complicate event characteristics. For more analysis about these issues, please refer to Sec. 5.3.

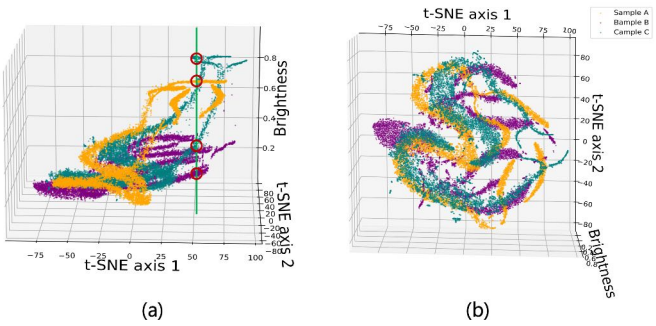


Figure 8: The ill-posedness of brightness estimation in the ISP process. We visualized the 5×5 region in the RAW image and the brightness of corresponding pixel in the color image at the center of this region. The results show that the same RAW region corresponds to different brightness levels in different images.

Summary: These sections show that the performance of numerous ISP methods on HVS sensor datasets varies significantly across different scenes. For instance, PyNet and AWNet (Dai et al., 2020) exhibit great variability between indoor and outdoor environments, underscoring that learning-based ISP methods are highly scene-dependent. This highlights the necessity for future work to analyze different scenes individually to fully understand the performance of a network. Furthermore, adding events to UNet significantly improves performance in outdoor scenarios but not indoors, mainly due to the flickering indoor lighting. Addressing this issue remains a crucial challenge for future research.

5.3 Discussion and Future Direction

Through the comprehensive and objective evaluation of various models on our dataset, we have also observed a number of findings that can bring insights for future work.

Significant Indoor-Outdoor Performance Gap on PyNet and AWNet (Dai et al., 2020): We observed a significant indoor-outdoor performance gap on PyNet (Ignatov et al., 2020b) and AWNet (Dai et al., 2020). PyNet performs better in outdoor scenes than indoor, ranking the top of all models, while AWNet (Dai et al., 2020) shows quite the opposite behavior. Generally, outdoor scenes have more dynamic and varied lighting compared to indoor environments, which are difficult for models to learn. The original AWNet (Dai et al., 2020) is designed to be trained in a multi-stage manner with different loss functions. Therefore it might have fallen into sub-optima when trained end-to-end in our experiment, resulting in the poor performance in modeling the harder outdoor scenes.

Local Brightness Artifacts: Artifacts occur when the brightness in certain image areas significantly deviates from the overall luminance (see Fig. 7). We investigated this by examining the relationship between a brightness of a pixel and the RAW data within its 5×5 vicinity. We treat the neighboring RAW data as a 25-dimensional vector, and apply t-SNE to project it onto a 2D plane, recording the (x, y) coordinates. We then converted the RGB values of the pixel to YUV, recording the Y (brightness) as the z coordinate, as shown in Fig. 8. By plotting pixels from three random images in 3D (Fig. 8), we show that pixel brightness and neighboring RAW data have a non-injective relationship. Multiple brightness levels can emerge from the same RAW data, indicating that **global information**, not just local RAW value, is essential for accurately determining pixel brightness to avoid local artifacts.

Event Gains: The integration of events in our dataset significantly enhances performance in outdoor scenes when comparing EV-UNET with UNet, primarily due to the additional motion information and dynamic range provided by the events. However, simple fusion does not fully exploit these characteristics, highlighting the need for more sophisticated designs in future research. Conversely, performance decreases in indoor scenes, primarily due to the flickering of artificial light sources.

Flickering Artificial Lighting: Under certain indoor scenarios, some artificial light source (Xu et al., 2023), *e.g.*LEDs, flicker because of the alternating current frequency. Given that the event frame rate of the sensor significantly exceeds the usual AC frequency (50 or 60 Hz), the flickering lighting introduces considerable fluctuations in the event data over time. The distributions and features of events in these conditions are completely different from that in the natural lighting conditions, and could result in the model's failure in restoring the images from RAW data.

6 Conclusion

In this work, we present the first events-RAW paired dataset for event-guided ISP research. The dataset consists of 3373 high quality high resolution RAW images and corresponding **pixel-level aligned** events. Subsequently, good RGB frames are generated by a controllable ISP pipeline we proposed. A comprehensive evaluation and analysis of existing learnable ISPs and a simple event-guided ISP method are conducted on our dataset. Based on this analysis, we summarize some key points and challenges for event-guided ISP.

We wish to emphasize the potential of event data in ISP processes again. Event cameras have a high dynamic range and high temporal resolution, which surpass the limits of human vision systems. In terms of dynamic range and temporal sampling, the information captured by event sensor is somehow a superset of that of human eye. Therefore, generating images perceptible to human vision is a matter of downward compatibility.

Limitations: Firstly, the scale of our dataset is relatively small, because the HVS sensor we use is still in the prototype stage and the associated hardware is cumbersome and exhibits low stability, which has raised the cost in data collection and thus a limited size dataset. And yet we are committed to expanding the dataset with more diverse real-world scenarios in future research. Secondly, our dataset has not thoroughly addressed the issue of flickering in artificial lighting caused by alternating current, especially in indoor scenarios. The flickering considerably impairs the performance of our method and further research should pay attention to this problem.

REFERENCES

Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1692–1700, 2018. 2

Alpsentek. Alpix-eiger product overview: https://alpsentek.com/product, 2024. URL https://alpsentek.com/product. Accessed: 2024-05-19. 2, 3, 4

Matthew Anderson, Ricardo Motta, Srinivasan Chandrasekar, and Michael Stokes. Proposal for a standard default color space for the internet—srgb. In *Color and imaging conference*, volume 4, pp. 238–245. Society of Imaging Science and Technology, 1996. 6

Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3703–3712, 2019. 3

Dominique Brunet, Edward R Vrscay, and Zhou Wang. On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, 21(4):1488–1499, 2011. 7, 9

Prashant Chaudhari, Franziska Schirrmacher, Andreas Maier, Christian Riess, and Thomas Köhler. Merging-isp: Multi-exposure high dynamic range image signal processing. In *DAGM German Conference on Pattern Recognition*, pp. 328–342. Springer, 2021. 22

Honggang Chen, Xiaohai He, Linbo Qing, Yuanyuan Wu, Chao Ren, Ray E Sheriff, and Ce Zhu. Real-world single image super-resolution: A brief review. *Information Fusion*, 79:124–145, 2022. 8

Matheus Henrique Marques da Silva, Jhessica Victoria Santos da Silva, Rodrigo Reis Arrais, Wladimir Barroso Guedes de Araújo Neto, Leonardo Tadeu Lopes, Guilherme Augusto Bileki, Iago Oliveira Lima, Lucas Borges Rondon, Bruno Melo de Souza, Mayara Costa Regazio, et al. Isp meets deep learning: A survey on deep learning methods for image signal processing. *arXiv preprint arXiv:2305.11994*, 2023a. 3

Matheus Henrique Marques da Silva, Jhessica Victoria Santos da Silva, Rodrigo Reis Arrais, Wladimir Barroso Guedes de Araújo Neto, Leonardo Tadeu Lopes, Guilherme Augusto Bileki, Iago Oliveira Lima, Lucas Borges Rondon, Bruno Melo de Souza, Mayara Costa Regazio, et al. Survey on software isp methods based on deep learning. *arXiv preprint arXiv:2305.11994*, 2023b. 7

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Bm3d image denoising with shape-adaptive principal component analysis. In *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009. 6

Linhui Dai, Xiaohong Liu, Chengqi Li, and Jun Chen. Awnet: Attentive wavelet network for image isp. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pp. 185–201. Springer, 2020. 3, 7, 8, 9, 10, 24

Mauricio Delbracio, Damien Kelly, Michael S Brown, and Peyman Milanfar. Mobile computational photography: A tour. *Annual review of vision science*, 7:571–604, 2021. 1

Graham D Finlayson, Michal Mackiewicz, and Anya Hurlbert. Color correction using root-polynomial regression. *IEEE Transactions on Image Processing*, 24(5):1460–1470, 2015. 6

Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020. 1, 3

Eiki Goto, Murat Dogru, Takashi Kojima, and Kazuo Tsubota. Computer-synthesis of an interference color chart of human tear lipid layer, by a colorimetric approach. *Investigative ophthalmology & visual science*, 44(11):4693–4697, 2003. 2

Keigo Hirakawa and Thomas W Parks. Joint demosaicing and denoising. *IEEE Transactions on Image Processing*, 15(8):2146–2157, 2006. 2

Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pp. 2366–2369. IEEE, 2010. 7, 9

Jingwei Huang, Zhili Chen, Duygu Ceylan, and Hailin Jin. 6-dof vr videos with a single 360-camera. In *2017 IEEE Virtual Reality (VR)*, pp. 37–44. IEEE, 2017. 1

Andrey Ignatov, Radu Timofte, Zhilu Zhang, Ming Liu, Haolin Wang, Wangmeng Zuo, Jiawei Zhang, Ruimao Zhang, Zhanglin Peng, Sijie Ren, et al. Aim 2020 challenge on learned image signal processing pipeline. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pp. 152–170. Springer, 2020a. 7, 22, 23

Andrey Ignatov, Luc Van Gool, and Radu Timofte. Replacing mobile camera isp with a single deep learning model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 536–537, 2020b. 3, 7, 10, 22, 23

Shantanu Ingle and Madhuri Phute. Tesla autopilot: semi autonomous driving, an uptick for future autonomy. *International Research Journal of Engineering and Technology*, 3(9):369–372, 2016. 1

Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning event-based motion deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3320–3329, 2020. 3

Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao. Turning frequency to resolution: Video super-resolution via event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7772–7781, 2021. 1, 3

Byung-Hoon Kim, Joonyoung Song, Jong Chul Ye, and JaeHyun Baek. Pynet-ca: enhanced pynet with channel attention for end-to-end mobile image signal processing. In *European Conference on Computer Vision*, pp. 202–212. Springer, 2020. 7, 8, 9, 22, 23

Chong-Min Kyung et al. *Theory and applications of smart cameras*. Springer, 2016. 1

Yong-Keun Lee and John M Powers. Comparison of cie lab, ciede 2000, and din 99 color differences between various shades of resin composites. *International Journal of Prosthodontics*, 18(2), 2005. 6

Zhaowen Li, Tingcun Wei, and Ran Zheng. Design of black level calibration system for cmos image sensor. In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, volume 10, pp. V10–643. IEEE, 2010. 2

Zhihao Li, Si Yi, and Zhan Ma. Rendering nighttime image via cascaded color and brightness compensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 897–905, 2022. 21

Guoqiang Liang, Kanghao Chen, Hangyu Li, Yunfan Lu, and Lin Wang. Towards robust event-guided low-light image enhancement: A large-scale real-world event-image dataset and novel approach. *arXiv preprint arXiv:2404.00834*, 2024. 1, 3

Jinxiu Liang, Yixin Yang, Boyu Li, Peiqi Duan, Yong Xu, and Boxin Shi. Coherent event guided low-light video enhancement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10615–10625, 2023. 3

Zhetong Liang, Jianrui Cai, Zisheng Cao, and Lei Zhang. Cameranet: A two-stage framework for effective camera isp learning. *IEEE Transactions on Image Processing*, 30:2248–2262, 2021. 2, 3, 7, 8, 24

Carl C Liebe, Edwin W Dennison, Bruce Hancock, Robert C Stirbl, and Bedabrata Pain. Active pixel sensor (aps) based star tracker. In *1998 IEEE Aerospace Conference Proceedings (Cat. No. 98TH8339)*, volume 1, pp. 119–127. IEEE, 1998. 1

Songnan Lin, Jiawei Zhang, Jinshan Pan, Zhe Jiang, Dongqing Zou, Yongtian Wang, Jing Chen, and Jimmy Ren. Learning event-driven video deblurring and interpolation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pp. 695–710. Springer, 2020. 3

Daikun Liu, Teng Wang, and Changyin Sun. Voxel-based multi-scale transformer network for event stream processing. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 8

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021. 3, 8

Yunfan Lu, Guoqiang Liang, and Lin Wang. Self-supervised learning of event-guided video frame interpolation for rolling shutter frames. *arXiv preprint arXiv:2306.15507*, 2023a. 1, 3

Yunfan Lu, Zipeng Wang, Minjie Liu, Hongjian Wang, and Lin Wang. Learning spatial-temporal implicit neural representations for event-guided video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1557–1567, 2023b. 1, 2, 3, 8

Yunfan Lu, Yijie Xu, Wenzong Ma, Weiyu Guo, and Hui Xiong. Event camera demosaicing via swin transformer and pixel-focus loss. *arXiv preprint arXiv:2404.02731*, 2024. 8, 24

M Ronnier Luo, Guihua Cui, and Bryan Rigg. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 26(5):340–350, 2001. 6

Marc Mahy, Luc Van Eycken, and André Oosterlinck. Evaluation of uniform color spaces developed after the adoption of cielab and cieluv. *Color Research & Application*, 19(2):105–121, 1994. 6

Jon S McElvain and Walter Gish. Camera color correction using two-dimensional transforms. In *Color and Imaging Conference*, volume 21, pp. 250–256. Society for Imaging Science and Technology, 2013. 2

Nico Messikommer, Stamatios Georgoulis, Daniel Gehrig, Stepan Tulyakov, Julius Erbach, Alfredo Bochicchio, Yuanyou Li, and Davide Scaramuzza. Multi-bracket high dynamic range imaging with event cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 547–557, 2022. 1

MIPI Challenge 2024. Mobile intelligent photography and imaging workshop 2024. https://mipi-challenge.org/MIPI2024/, 2024. 2, 3

Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE international conference on computer vision*, pp. 261–270, 2017. 3

Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6820–6829, 2019. 3

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 7

T-C Poon and Partha P Banerjee. *Contemporary optical image processing with MATLAB*. Elsevier, 2001. 2

Rainbow-Johnny-Johnny-Image-Processing-Lim. Quadbayer cfa modified gradient-based demosaicing, 2022. URL https://www.mathworks.com/matlabcentral/fileexchange/116085-quadbayer-cfa-modified-gradient-based-demosaicing. Accessed: 2024-06-01. 6

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015. 2, 3, 8, 24

Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77:157–173, 2008. 4, 6, 19, 22

Cedric Scheerlinck, Henri Rebecq, Timo Stoffregen, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Ced: Color event camera dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019. 2, 3

Eli Schwartz, Raja Giryes, and Alex M Bronstein. Deepisp: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing*, 28(2):912–923, 2018. 3, 21

Waseem Shariff, Mehdi Sefidgar Dilmaghani, Paul Kielty, Mohamed Moustafa, Joe Lemley, and Peter Corcoran. Event cameras in automotive sensing: A review. *IEEE Access*, 2024. 1, 3

Ardhendu Shekhar Tripathi, Martin Danelljan, Samarth Shukla, Radu Timofte, and Luc Van Gool. Transform your smartphone into a dslr camera: Learning the isp in the wild. In *European Conference on Computer Vision*, pp. 625–641. Springer, 2022. 2, 21

Chen Song, Qixing Huang, and Chandrajit Bajaj. E-cir: Event-enhanced continuous intensity recovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7803–7812, 2022. 3

Gui Yun Tian, Duke Gledhill, David Taylor, and David Clarke. Colour correction for panoramic imaging. In *Proceedings Sixth International Conference on Information Visualisation*, pp. 483–488. IEEE, 2002. 4

Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. Time lens: Event-based video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16155–16164, 2021. 1, 3

Stepan Tulyakov, Alfredo Bochicchio, Daniel Gehrig, Stamatios Georgoulis, Yuanyou Li, and Davide Scaramuzza. Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17755–17764, 2022. 1, 3

Bishan Wang, Jingwei He, Lei Yu, Gui-Song Xia, and Wen Yang. Event enhanced high-quality image recovery. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pp. 155–171. Springer, 2020a. 8

Wencheng Wang, Xiaojin Wu, Xiaohui Yuan, and Zairui Gao. An experiment-based review of low-light image enhancement methods. *Ieee Access*, 8:87884–87917, 2020b. 1

Ching-Chih Weng, Homer Chen, and Chiou-Shann Fuh. A novel automatic white balance method for digital still cameras. In *2005 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3801–3804. IEEE, 2005. 2

Li Xiaopeng, Zeng Zhaoyuan, Fan Cien, Zhao Chen, Deng Lei, and Yu Lei. Hdr imaging for dynamic scenes with events. *arXiv preprint arXiv:2404.03210*, 2024. 1

Yazhou Xing, Zian Qian, and Qifeng Chen. Invertible image signal processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6287–6296, 2021. 7, 8, 22, 23

Fang Xu, Lei Yu, Bishan Wang, Wen Yang, Gui-Song Xia, Xu Jia, Zhendong Qiao, and Jianzhuang Liu. Motion deblurring with real events. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2583–2592, 2021. 3

Lexuan Xu, Guang Hua, Haijian Zhang, Lei Yu, and Ning Qiao. ” seeing” electric network frequency from events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18022–18031, 2023. 10

Lexuan Xu, Guang Hua, Haijian Zhang, and Lei Yu. “seeing” enf from neuromorphic events: Modeling and robust estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024a. 26

Senyan Xu, Zhijing Sun, Jiaying Zhu, Yurui Zhu, Xueyang Fu, and Zheng-Jun Zha. Demosaicformer: Coarse-to-fine demosaicing network for hybridevs camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1126–1135, 2024b. 21

Qingyu Yang, Guang Yang, Jun Jiang, Chongyi Li, Ruicheng Feng, Shangchen Zhou, Wenxiu Sun, Qingpeng Zhu, Chen Change Loy, Jinwei Gu, et al. Mipi 2022 challenge on quad-bayer re-mosaic: Dataset and report. In *European Conference on Computer Vision*, pp. 21–35. Springer, 2022. 2

Wu Yaqi, Fan Zhihao, Chu Xiaofeng, Ren Jimmy S., Li Xiaoming, Yue Zongsheng, Li Chongyi, Zhou Shangcheng, Feng Ruicheng, Dai Yuekun, Yang Peiqing, Loy Chen Change, et al. Mipi 2024 challenge on demosaic for hybridevs camera: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2, 3

Ke Yu, Zexian Li, Yue Peng, Chen Change Loy, and Jinwei Gu. Reconfigisp: Reconfigurable camera image processing pipeline. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4248–4257, 2021. 21

Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum. Image deblurring with blurred/noisy image pairs. In *ACM SIGGRAPH 2007 papers*, pp. 1–es. 2007. 1

LU Yunfan, Guoqiang Liang, and Lin Wang. Uniinr: Unifying spatial-temporal inr for rs video correction, deblur, and interpolation with an event camera. 2023. 1, 3

Kaihao Zhang, Wenqi Ren, Wenhan Luo, Wei-Sheng Lai, Björn Stenger, Ming-Hsuan Yang, and Hongdong Li. Deep image deblurring: A survey. *International Journal of Computer Vision*, 130(9):2103–2130, 2022. 1, 8

Lin Zhang, Ying Shen, and Hongyu Li. Vsi: A visual saliency-induced index for perceptual image quality assessment. *IEEE Transactions on Image processing*, 23(10):4270–4281, 2014. 24

Lin Zhang, Lei Zhang, and Alan C Bovik. A feature-enriched completely blind image quality evaluator. *IEEE Transactions on Image Processing*, 24(8):2579–2591, 2015. 24

Xiang Zhang and Lei Yu. Unifying motion deblurring and frame interpolation with events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17765–17774, 2022. 3

Xinyu Zhou, Peiqi Duan, Yi Ma, and Boxin Shi. Evunroll: Neuromorphic events based rolling shutter image correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17775–17784, 2022. 1

Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018. 1

Appendix / Supplemental Material

Due to space limitations in the main text, additional details are presented in the supplementary material. Specifically:

- **Sec. A**: We provide more information about the imaging principle, process and potential of the sensor.
- **Sec. B**: We provide further details on dataset collection.
- **Sec. C**: We delve into more specifics about the controllable ISP.
- **Sec. D**: We describe the characteristics of the methods compared in the Benchmark.
- **Sec. E**: We provide more metrics of the methods compared in the Benchmark.
- **Sec. F**: We discuss additional points of discussion.

A Hybrid Sensor Imaging Process, Principles and Potential

This section describes the imaging process and working principles of the hybrid vision sensor (HVS) used in this paper. The HVS combines quad Bayer pattern-based RGB imaging with event-based sensing, enabling high temporal resolution and high dynamic range. The following subsections elaborate on the Quad Bayer structure, event generation principles, and rolling shutter readout process.

A.1 Quad Bayer Pattern and RGB Imaging

The hybrid sensor utilizes a quad Bayer pattern, as shown in Fig. 1, where each group of four pixels consists of three color pixels (red, green, and blue) and one event pixel. Let I_{RAW} represent the RAW image captured by the sensor, with pixel intensity values denoted as $I_{RAW}(x, y)$. For the RGB pixels, the values correspond to the photonic response of the sensor to incoming light, represented by:

$$I_{RGB}(x, y) = K_{RGB} \cdot \Phi(x, y) + N_{RGB}, \tag{1}$$

where $\Phi(x, y)$ is the incident light intensity, K_{RGB} is the sensitivity coefficient, and N_{RGB} is the noise term.

The quad Bayer pattern increases the effective resolution of the sensor by allowing demosaicing algorithms to interpolate missing color information. Additionally, the rolling shutter mechanism is used to sequentially expose rows of the sensor, resulting in temporal offsets across the frame. This is illustrated in Fig. 10.

A.2 Event Generation Principles

In addition to RGB imaging, the hybrid sensor includes event pixels that detect changes in luminance. These event pixels operate in the logarithmic domain and generate an event $E(x, y, t, p)$ whenever the change in logarithmic intensity exceeds a predefined threshold θ. The mathematical model for event generation is as follows:

$$\Delta L(x, y, t) = \log(I(x, y, t)) - \log(I(x, y, t - \Delta t)), \tag{2}$$

$$E(x, y, t, p) = \begin{cases} 1, & \text{if } \Delta L(x, y, t) > \theta, \\ -1, & \text{if } \Delta L(x, y, t) < -\theta, \end{cases} \tag{3}$$

where $I(x, y, t)$ is the light intensity at pixel (x, y) and time t, Δt is the sampling interval, and $p \in \{-1, 1\}$ represents the polarity of the event (indicating an increase or decrease in luminance)

High Temporal Resolution: The event generation process is asynchronous and occurs independently at each pixel, triggered only when a significant luminance change is detected. This enables extremely high temporal resolution, as events can be recorded at microsecond-scale intervals. Let $f_{temporal}$ represent the temporal resolution of event recording, which depends on the sampling interval Δt:

$$f_{temporal} = \frac{1}{\Delta t}. \tag{4}$$

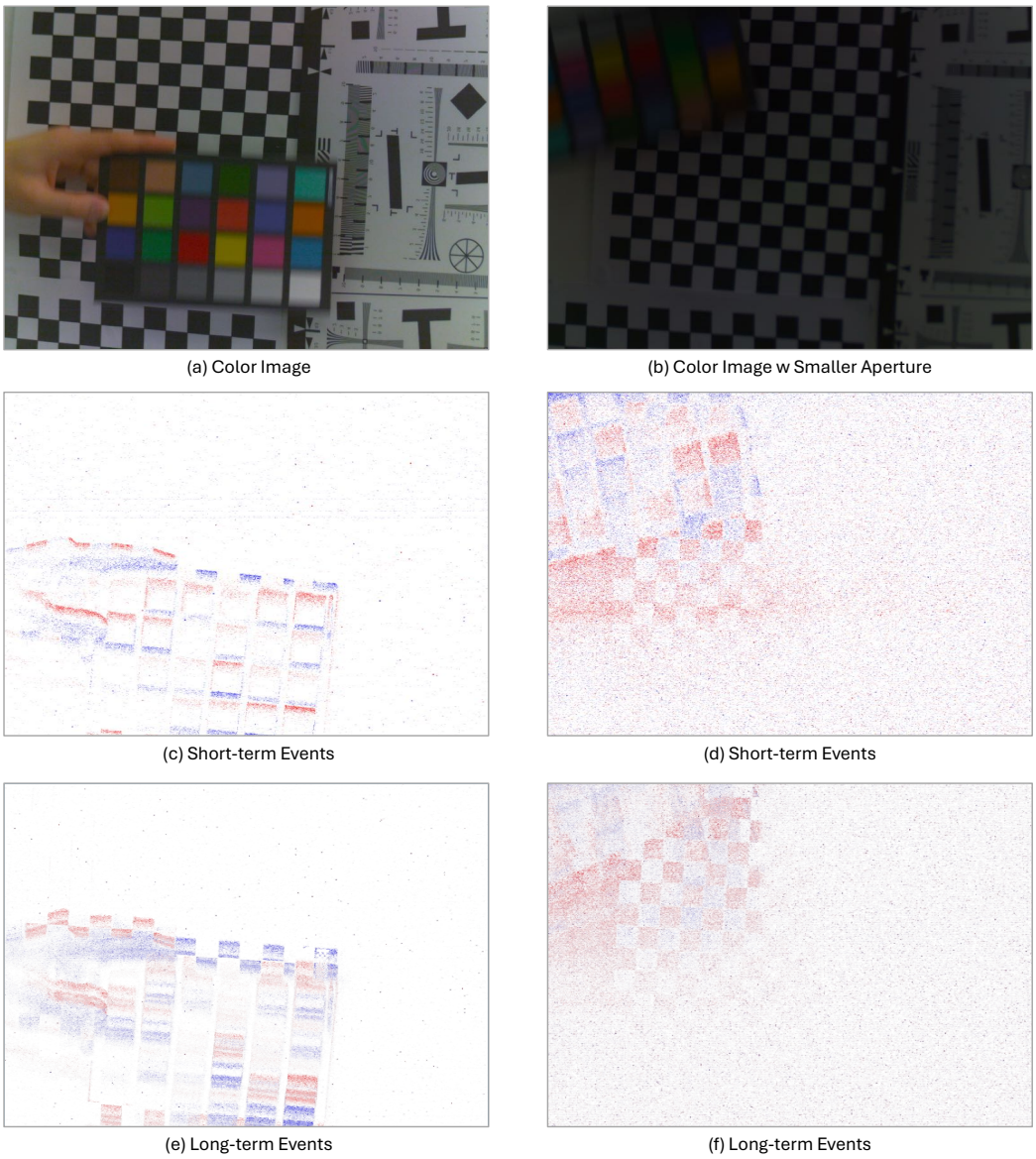


Figure 9: In fast-motion scenarios and low-light conditions, real-world data demonstrates the advantages of events in achieving higher temporal resolution and greater dynamic range. For example, in the fast-motion scene, the color checker in (a) exhibits blurred edges, while the corresponding short-term event frame shows sharper edges, capturing motion more accurately. In low-light conditions, (b) illustrates the limitations of traditional RGB imaging, where details are lost due to insufficient lighting. However, as shown in (d) and (f), the event data captures motion effectively even under low light. Nevertheless, it is also evident that events exhibit increased noise levels in low-light conditions, as highlighted in (f). Our data opens the possibility for future low-light enhancement and deblurring in the RAW domain via events.

This high temporal resolution allows the sensor to effectively capture rapid motion and high-speed dynamics that traditional RGB cameras, limited by frame rates, cannot resolve. For example, a rolling ball or moving object creates a continuous stream of events corresponding to pixel-level luminance changes, enabling precise tracking of motion trajectories in real-time. This characteristic is particularly advantageous for motion deblurring and temporal interpolation tasks.

High Dynamic Range (HDR): The logarithmic domain operation of the event pixels inherently provides a high dynamic range. Unlike traditional RGB sensors, which saturate under bright light-

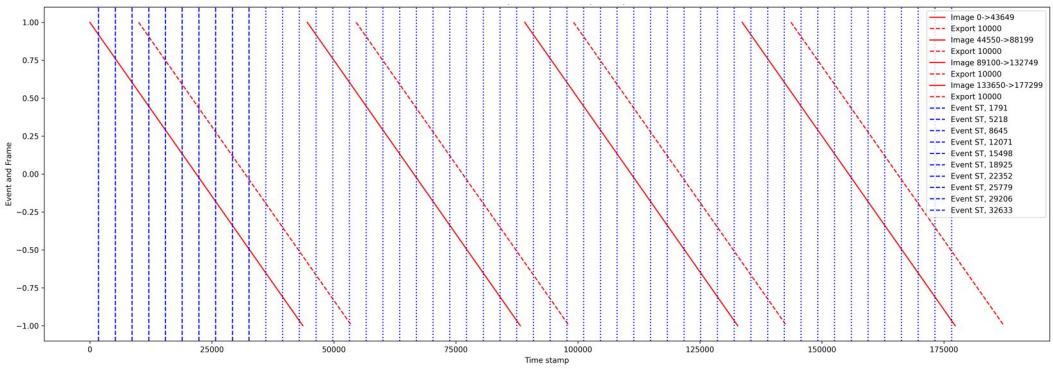


Figure 10: The camera utilized for data collection features a rolling shutter mode. The horizontal axis in the figure represents time, and the vertical axis represents each row. Our sensor outputs both APS frames and EVS events concurrently, with both being aligned in space and time. Specifically, the APS frames are captured using a rolling shutter exposure method. In the diagram, the red lines represent the rolling shutter pattern; the solid red line corresponds to the start of the rolling shutter exposure, and the dashed line signifies the end of the exposure.The blue straight line represents events.

ing conditions or lose detail in shadows, event pixels respond to relative changes in luminance rather than absolute intensity. Let I_{max} and I_{min} represent the maximum and minimum detectable intensities, respectively. The dynamic range DR can be expressed in decibels (dB) as:

$$DR = 20 \log_{10} \left(\frac{I_{max}}{I_{min}} \right). \tag{5}$$

Because events are triggered by logarithmic intensity changes, the sensor is capable of detecting changes over a wide range of luminance levels, from very dark to extremely bright conditions. This property enables effective operation in scenarios with challenging lighting conditions, such as low-light environments or scenes with high contrast between bright and shadowed regions.

Practical Implications: The combination of high temporal resolution and high dynamic range makes the hybrid sensor particularly well-suited for applications involving fast motion or extreme lighting conditions. As illustrated in Fig. 9, events accurately capture motion details even in low-light scenarios while preserving high-frequency temporal information. These unique characteristics complement the RGB output, enhancing the performance of hybrid sensor ISP tasks such as motion deblurring, HDR reconstruction, and low-light enhancement.

A.3 Rolling Shutter and Temporal Alignment

The RGB output of the hybrid sensor follows a rolling shutter exposure mechanism, as shown in Figure 10. In this method, each row of the sensor is exposed sequentially, introducing temporal offsets between rows. Let $t_{start}(r)$ and $t_{end}(r)$ represent the start and end times of exposure for row r, respectively. The effective exposure time for row r is given by:

$$T_{exp}(r) = t_{end}(r) - t_{start}(r). \tag{6}$$

To achieve temporal alignment between the RGB and event streams, the event data are synchronized with the rolling shutter exposure times. Both events and frames have unified timestamps to ensure time alignment. This alignment is critical for event-guided ISP tasks, where temporal information from events complements the spatial information in RGB frames.

A.4 Potential Benefits of Events for Different Tasks in ISP

Demosaicing: *Task Objective:* Reconstruct the full-resolution RGB image $I_{RGB}(x, y)$ from incomplete color samples in the quad Bayer pattern. *Benefits of Events:* High temporal resolution events provide precise edge information via spatial gradients $\nabla E(x, y, t)$:

$$\nabla E(x, y, t) = \left(\frac{\partial E}{\partial x}, \frac{\partial E}{\partial y} \right),$$

guiding interpolation along edges and reducing artifacts like color fringing. The high dynamic range aids in preserving details across varying luminance levels.

Denoising: *Task Objective:* Reduce noise N_{RGB} in RAW image $I_{RAW}(x, y)$ to enhance image quality. *Benefits of Events:* Events, triggered by significant luminance changes $\Delta L(x, y, t) > \theta$, help differentiate signal from noise. By weighting the denoising process with event activity $E(x, y, t)$:

$$I_{\text{denoised}}(x, y) = I_{RAW}(x, y) - w(x, y) \cdot N_{RGB}(x, y),$$

where $w(x, y) = f(E(x, y, t))$, we suppress noise while preserving details in dynamic regions.

White Balancing: *Task Objective:* Adjust image colors to render neutral whites under varying illumination. *Benefits of Events:* Using event rate $r_E(x, y) = \frac{\Delta E}{\Delta t}$, we detect illumination changes and adjust white balance coefficients K_{WB}:

$$I_{\text{WB}}(x, y) = K_{\text{WB}} \cdot I_{RAW}(x, y),$$

enabling real-time adaptation to lighting variations due to the high dynamic range and temporal resolution of events.

Color Correction: *Task Objective:* Map image colors to a standard color space for accurate representation. *Benefits of Events:* Events highlight regions with significant luminance shifts, indicating potential color deviations. Incorporating event information into the color correction matrix $\mathbf{M}_{\text{CC}}(E)$:

$$I_{\text{corrected}}(x, y) = \mathbf{M}_{\text{CC}}(E(x, y, t)) \cdot I_{\text{WB}}(x, y),$$

allows dynamic adjustment for scene changes, enhancing color fidelity, especially in scenes with rapid motion or high contrast.

In summary, the hybrid sensor enables simultaneous RGB and event data acquisition, leveraging the strengths of both modalities. The RGB data provide high spatial resolution, while the event data capture motion and high-frequency changes with low latency and high dynamic range. This unique combination not only enhances the traditional ISP process but also opens up significant potential for advanced imaging applications. This combination facilitates advanced imaging tasks, including motion deblurring, and low-light enhancement in future, as show in Fig. 12.

B More Details about Dataset Collection

In Fig. 11, we present additional samples from our dataset, showcasing the diversity and richness of the collected data. The figure includes examples of RAW images, their corresponding high-quality RGB frames, and the associated event streams. These examples highlight the variety of scenes captured, encompassing urban environments with buildings, natural landscapes with vegetation, intricate textures, vibrant flowers, and more.

To ensure accurate color representation, each scene includes an image featuring a color card, which is systematically used for color calibration and estimation during the dataset processing. This approach enhances the reliability and usability of the dataset for ISP tasks.

By including diverse real-world scenarios, our dataset provides a robust platform for training and benchmarking ISP algorithms, particularly in the context of event-guided approaches.

In Fig. 12, we present additional data collection scenarios, encompassing various scenes and different weather conditions. We used LabelMe (Russell et al., 2008) to annotate the positions of the ColorChecker, specifically marking four points: cyan, white, brown, and black. Fig. 13 shows our annotation interface. All annotated location information is stored in JSON format and forms a one-to-one correspondence with the image.

B.1 Discussion on Dataset Scale

To further demonstrate the adequacy of our dataset, we provide a comparative analysis with the most related datasets in Table 5. Our dataset contains 3,373 images with a resolution of 2248×3264 pixels (approximately 7.3 million pixels per image). Compared to MIPI, which includes only 800 images

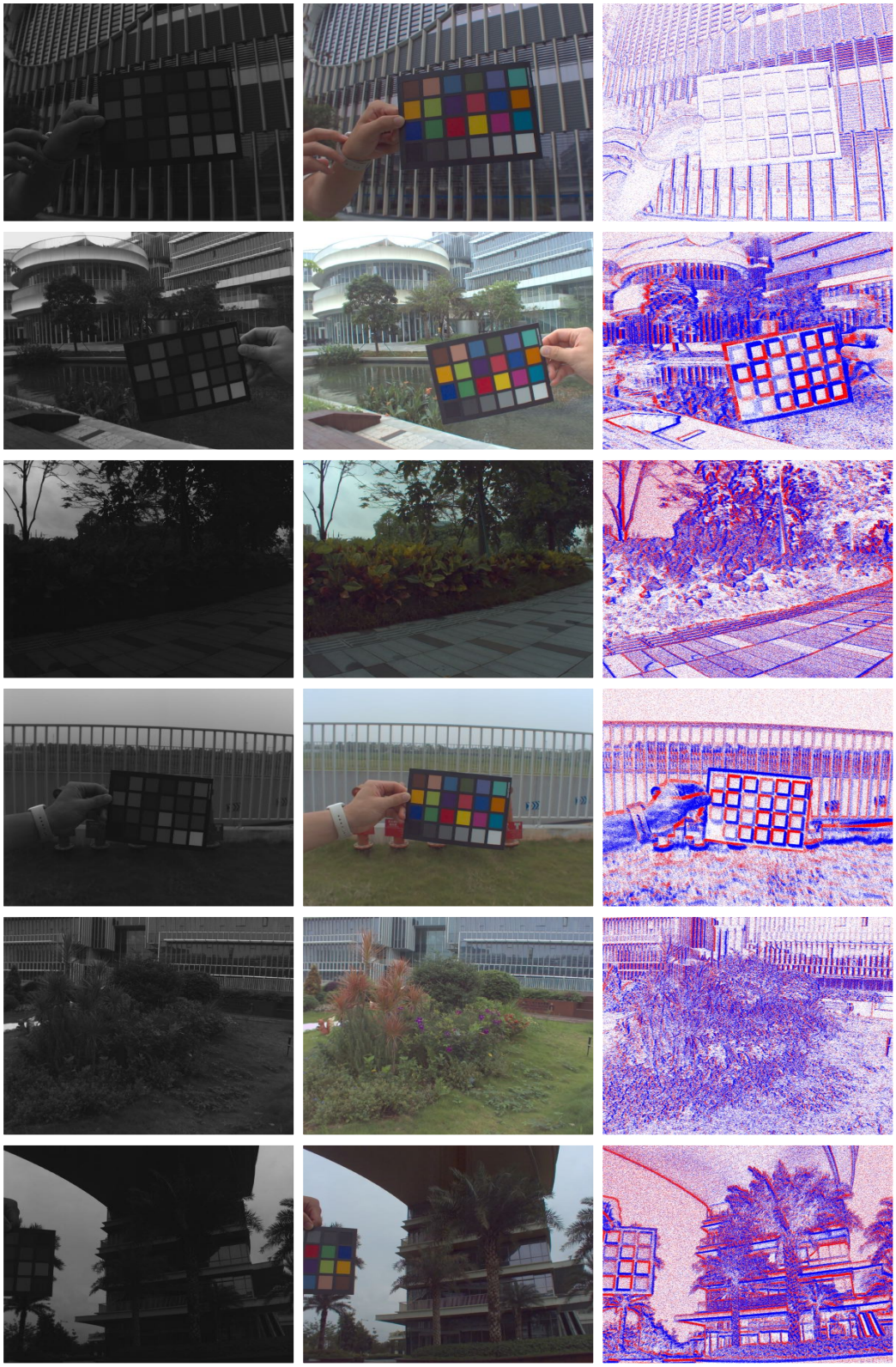


Figure 11: More samples in our dataset, from left to right: RAW, Good RGB frame and corresponding events..

Table 5: Size Comparison of Related Datasets. The resolution of MIPI datasets is not uniform.

Dataset	Resolution	Scale	Real-World	Events	Tasks	Publication
Ours	2248×3264	3373	Yes	Yes	Hybrid Sensor ISP	-
MIPI	2040×1356	800	No	No	Hybrid Sensor ISP	CVPR 2024
ISPW	1368×1824	197	Yes	No	ISP	ECCV 2022
NR2R	3464×5202	150	Yes	No	ISP	CVPR 2022
DeepISP	3024×4032	110	Yes	No	ISP	IEEE TIP 2018

with resolutions around $2K$ (e.g., 2040×1356), our dataset is over four times larger and is based on real-world data rather than simulated data. This difference makes our dataset more representative and applicable for real-world ISP tasks.

It is worth noting that the MIPI dataset, despite its smaller scale, has already been demonstrated to support the training of large networks such as Transformers (Xu et al., 2024b). Therefore, our larger dataset is even better suited for training and testing ISP models, offering greater potential for comprehensive research.

In addition to MIPI, we also consider prior ISP datasets such as ISPW (Shekhar Tripathi et al., 2022), which contains 197 groups of images, some of which have higher resolutions (e.g., 4480×6720). NR2R (Li et al., 2022) and DeepISP (Schwartz et al., 2018) as traditional ISP datasets have no more than 200 images for training. But both have high resolution. However, the total image count in ISPW is significantly smaller than our dataset, and it does not incorporate event data.

The scale of our dataset not only ensures a sufficient number of samples but also provides high-resolution data, enabling effective training and testing for ISP tasks. Additionally, the dataset includes diverse scenes, lighting conditions, and event streams, which further enhance its applicability to hybrid sensor ISP research.

In summary, our dataset provides a comprehensive resource for ISP research. Its real-world nature distinguishes it from existing datasets and makes it particularly well-suited for event-guided ISP tasks. Moreover, we are committed to the long-term maintenance of this dataset and plan to expand it in the future to accommodate larger and more complex tasks.

C More Details about Controllable ISP

The MATLAB demo code of the Controllable ISP is provided in the end of the appendix.In this section, we will elucidate further details in comment.

Fixed Pattern Noise: Practically, we capture pure black images (with the lens cap on) using different exposure times. The black frames captured with identical exposure times are averaged to obtain the fixed pattern noise. This noise indicates the potential deviations of some pixels, deviations which, if not addressed, can be exacerbated. The physical meaning of this noise is that even without optics, these pixels will have intensity output due to dark current.

Sensor Value Range: Typically in the hardware design of sensors, the green channel will obtain a larger value than the red and blue channels, as shown in Fig. 14. Therefore, when capturing a cloudy sky (which appears white to the human eye), the green channel may reach its maximum value due to overexposure, while the red and blue channels do not, resulting in a pinkish hue in the sky. In such cases, we artificially set the overexposure. The specific operation is to use 95% of the preset value of the sensor when normalizing the 8-bit values, setting the maximum value to 242 (255×0.95) for normalization.

D More Details about Benchmark Methods

In this section we explain in more detail the methods of the benchmark in the main paper.

(1) Full Pipeline ISP employs CNN architectures to streamline traditional ISP processes such as demosaicing, white balancing, and denoising, enabling a direct conversion from RAW images to RGB outputs in an end-to-end manner. This innovative approach has catalyzed extensive research, leading to the development of sophisticated models such as ReconfigISP (Yu et al., 2021), Merging-

Figure 12: More dataset scenes and labels. Our dataset encompasses various indoor and outdoor scenes, captured under different weather conditions, including cloudy and sunny days. The initial segments of our videos include a ColorChecker, which has been annotated using LabelMe (Russell et al., 2008).

ISP (Chaudhari et al., 2021), PyNet (Ignatov et al., 2020b), PyNetCA (Kim et al., 2020), InvertISP (Xing et al., 2021), and MV-ISPNet (Ignatov et al., 2020a). In our benchmark, we selected several models from a reputable open-source paper for comprehensive evaluation. Specifically, we chose MV-ISPNet, which secured first place at AIM 2020, demonstrating its robustness. Alongside, we included PyNet and its enhanced variant, PyNetCA, which incorporates attention layers for more in-depth analysis. Additionally, we incorporated InvertISP, known for its proven ability to adeptly handle various scenarios.

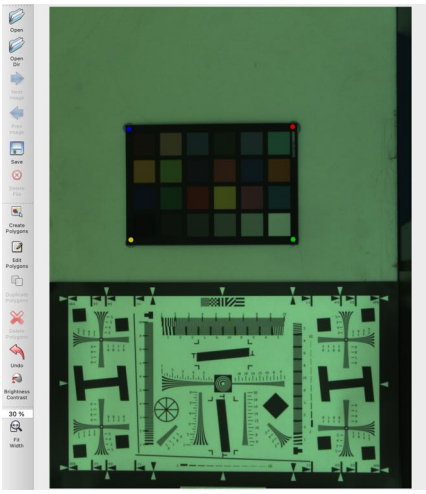


Figure 13: LabelMe annotation interface: We use LabelMe software to mark the four corners of a color card in images, which are white, cyan, brown, and black respectively. In practice, the images annotated by LabelMe are demosaiced from RAW format. While annotators can distinguish the colors, there are deviations in the image's inherent color representation.

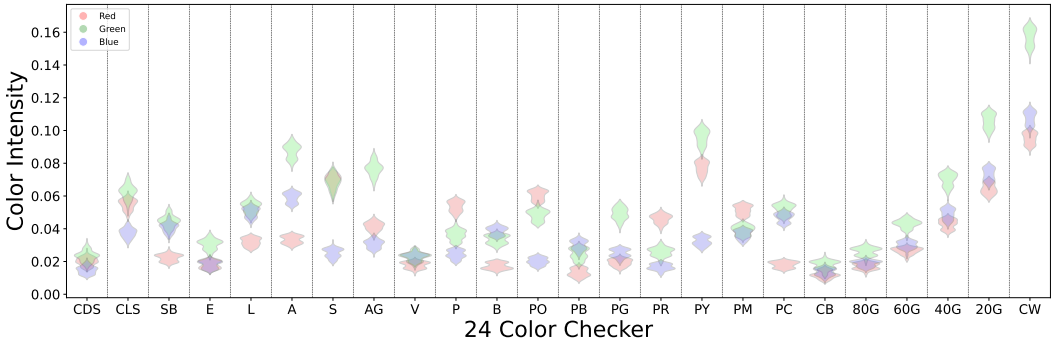


Figure 14: The distribution of colors on the color card in different frames under the same light source.

InvertISP (Xing et al., 2021): InvertISP is a pipeline with a specially designed reversible structure for both rendering RGB images from RAW and to inversely recover the RAW data from RGB images. It uses a series of reversible affine coupling layers and 1×1 convolutional layers to build a single reversible neural network that can map from RAW data to sRGB data, and can inversely restore RAW data from compressed RGB images.

MV-ISPNet (Ignatov et al., 2020a): MV-ISPNet is a multi-level wavelet ISP network based on U-Net. It takes advantage of the Multi-level Wavelet CNN (MWCNN) and Residual Channel Attention Network (RCAN) architectures, minimizes information loss through residual groups and discrete wavelet transforms, and combines multiple loss functions and self-integration methods to improve image quality.

PyNet (Ignatov et al., 2020b): PyNet utilizes a stack of CNN layers with different resolution level to process the image, which allows the network to learn a more diverse set of features, ranging from global brightness / color to local texture enhancement.

PyNetCA (Kim et al., 2020): PyNetCA is an enhanced version of the original PyNet. It adopts an inverted pyramid structure and considers both global and local features of the image through multi-scale feature fusion and residual connection. With a channel attention module (CA) to emphasize important channel features, and uses a sub-pixel reconstruction module (SRM) in the last layer to

improve upsampling efficiency and image quality through 1×1 convolution and sub-pixel shuffling technology.

(2) Stage-wise ISPs:.Instead of replacing the entire ISP pipeline with a single neural network, stage-wise learning based ISPs employ multiple specifically designed modules to perform different sub tasks and organize them sequentially or in parallel order to generate the final output. Note that these modules are sometimes trained independently in some models, in our experiment we only keep the model structures and train them end-to-end.

CameraNet (Liang et al., 2021): CameraNet designs two sequential modules and trains them separately to perform different tasks. The Restore-Net component is trained for demosaicing, white balancing and denoising while the Enhance-Net for sRGB gamma mapping and detail adjustments. The Pytorch version of CameraNet is not available, and therefore we experiments on a converted version.

AWNet (Dai et al., 2020): AWNet employs two parallel UNet-based modules to capture global and local content. The modules take in the original RAW image and a pseudo-demosaiced image generated from the RAW image, then the output of these two modules are averaged to produce the final output.

(3) Image Enhancement Network Based ISPs:. Transformer-based models have demonstrated high capabilities in image enhancement tasks (e.g., deblurring, super-resolution). Even though these models are not specifically designed for ISP, a minor conversion (i.e. replacing the projector in the output layer) could evoke their potentials in ISP tasks.

UNet (Ronneberger et al., 2015): UNet is a CNN-based model which has been widely adopted in areas of image processing due to its high performance in dealing with various sizes of images and modelling complex structures within them. In our experiment a UNet is trained to perform the task of ISP.

Swin Transformer (Lu et al., 2024): Swin Transformer is a transformer based general-purpose backbone for image processing. It produces a hierarchical representation with shifted windows transformer blocks and brings greater efficiency by limiting self-attention computation to non-overlapping local windows while also allowing for cross-window connection.

E BENCHMARKING METRICS

In addition to PSNR and SSIM, the inclusion of Natural Image Quality Evaluator (NIQE) (Zhang et al., 2015) and Perceptual Index (PI) (Zhang et al., 2014) provides deeper insights into the perceptual quality and naturalness of images across various models, as shown in Table. 6, and Table. 7. The ablation study analysis of the methods in these two tables, including a discussion of the best performances, a comparison between event-based methods (eSL and EV-UNet) and pure RGB methods, and the differences of these methods in indoor and outdoor scenes.

Best Performance Analysis: ***Indoor Scenes (Table. 6)***: Swin-Transformer exhibited excellent performance in indoor scenes, achieving the lowest average NIQE (7.7104) and PI (7.2125) values. This indicates its advantage in enhancing both image quality and perceptual quality. InvertISP closely followed, with an average NIQE of 8.8646 and PI of 7.8543, demonstrating good natural image quality. AWNet also showed balanced performance, with an average NIQE of 8.5311 and PI of 7.9843. ***Outdoor Scenes (Table 7.)***: InvertISP performed the best in outdoor scenes, achieving the lowest average NIQE (6.7187) and a relatively low PI (6.8720), especially excelling in complex flower and building scenes. Swin-Transformer also demonstrated excellent performance in outdoor scenes, with an average NIQE of 7.0284 and PI of 7.2255. eSL performed well in outdoor scenes, achieving an average NIQE of 6.9509 and PI of 7.1445.

Comparison Between Event-Based Methods and Pure RGB Methods: eSL achieved good NIQE and PI values in both indoor and outdoor scenes. In particular, in outdoor scenes, it obtained an average NIQE of 6.9509 and PI of 7.1445, which are close to the best performances. EV-UNet's performance was relatively average in both types of scenes. It had an average indoor NIQE of 10.6250 and PI of 9.2131; in outdoor scenes, it achieved an average NIQE of 8.2537 and PI of 7.8705. Pure RGB Methods Swin-Transformer, as a pure RGB method, performed excellently in both indoor and outdoor scenes. This indicates its good generalization ability when handling different scenes.

Table 6: Comparison of Methods on HVS ISP Dataset indoor scenes. * refer to the results obtained by the same model with different hyperparameters.

	1-In-Fruit-2		3-In-ColChecker-40		4-In-RLChart-10		Average	
Methods	**NIQE**	**PI**	**NIQE**	**PI**	**NIQE**	**PI**	**NIQE**	**PI**
PyNET	10.5604	8.9322	8.6510	7.8808	8.6960	7.9034	9.3024	8.2388
PyNET*	10.6142	9.0591	9.5254	8.4616	9.2743	8.2574	9.8046	8.5927
PyNetCA	9.5175	8.4227	9.5633	8.3391	10.0280	8.4848	9.7029	8.4155
InvertISP	9.6301	8.2957	7.7260	7.2639	9.2377	8.0033	8.8646	7.8543
MV-ISPNet	10.8919	9.2025	9.6756	8.3477	9.3904	8.3674	9.9859	8.6392
CameraNet	771.8745	392.2881	771.8748	392.2880	771.8735	392.2873	771.8743	392.2878
AWNet	9.0410	8.4831	8.8963	8.2489	7.6562	7.2208	8.5311	7.9843
Swin-Transformer	7.8467	7.3106	7.9101	7.2932	7.3744	7.0336	7.7104	7.2125
UNet	10.7405	9.3207	9.7168	8.6760	9.5778	8.5257	10.0117	8.8408
UNet*	11.5525	9.9720	10.3320	9.0237	10.7163	9.2440	10.8669	9.4132
eSL	8.9799	8.1130	8.5955	7.8486	8.9683	8.0099	8.8479	7.9905
EV-UNet	10.3218	9.2965	10.7447	9.1840	10.8084	9.1588	10.6250	9.2131

InvertISP, although a pure RGB method, performed outstandingly in outdoor scenes, especially in improving image quality under complex lighting conditions. The event-based method eSL's performance in outdoor scenes was close to the best, which may be due to the advantage of event data in capturing dynamic and high dynamic range scenes. EV-UNet's performance was slightly inferior to eSL, which may be related to its model structure or the degree to which it utilizes event data. Pure RGB methods like Swin-Transformer and InvertISP performed outstandingly, indicating that even without event data, excellent performance can be achieved through improved model structures and algorithms.

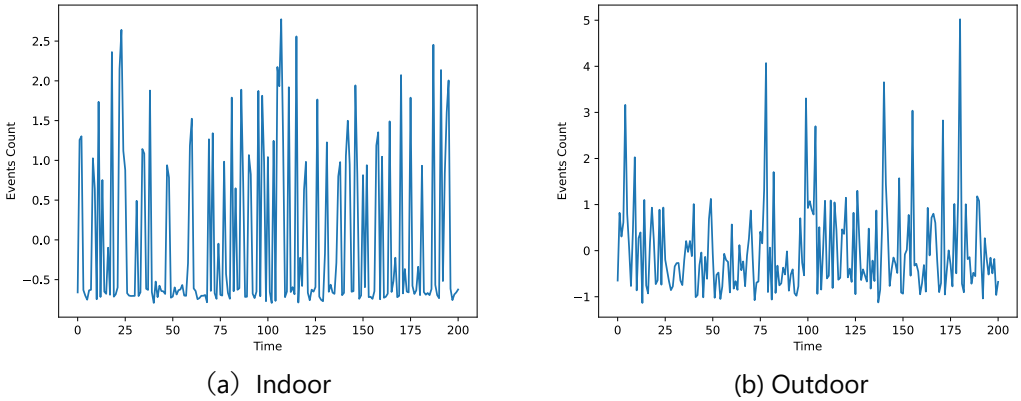
Differences Between Indoor and Outdoor Scenes: Performance Differences: Most methods exhibit better NIQE and PI values in outdoor scenes than in indoor scenes. This may be because outdoor scenes have more complex lighting conditions and content, providing more information to the models. Advantages of Event-Based Methods: Event-based methods display more significant advantages in outdoor scenes, especially when handling rapid changes and high dynamic range environments. In such cases, event data can provide additional information to improve image quality. Model Generalization Ability: Models like Swin-Transformer maintain consistently high performance in both indoor and outdoor scenes, demonstrating good generalization ability suitable for various environments.

Impact of Hyperparameters on the Model: We observed that hyperparameters can have a impact on model performance. For instance, adjustments to learning rate, batch size, or the choice of optimizer often lead to measurable variations in results. As a benchmark study, we strive to ensure fair and unbiased evaluation across all methods by carefully tuning hyperparameters to achieve reasonable performance. However, finding the optimal hyperparameters for each model remains a challenging task, particularly given the computational costs and the inherent differences in how models respond to tuning. In practice, hyperparameter tuning often requires balancing empirical results with theoretical insights, as exhaustive grid searches are rarely feasible. Despite these challenges, we will provide all training codes to ensure transparency and reproducibility, while acknowledging that further fine-tuning might yield even better results for some models.

In summary, the best-performing models vary across different scenes; however, Swin-Transformer and InvertISP demonstrate excellent performance in both indoor and outdoor environments. The event-based method eSL performs close to the best in outdoor scenes, confirming the effectiveness of event data in complex scenes. Pure RGB methods can also achieve excellent performance by improving model structures. However, in specific scenes, the introduction of event data may provide additional advantages. The performance differences of methods in indoor and outdoor scenes suggest that model design and training need to consider scene characteristics to achieve the best results.

Table 7: Comparison of methods on HVS ISP dataset outdoor scenes. * refer to the results obtained by the same model with different hyperparameters.

	2-OUT-Tree-2		3-Out-Flower-2		4-Out-Building-1		Average	
Methods	**NIQE**	**PI**	**NIQE**	**PI**	**NIQE**	**PI**	**NIQE**	**PI**
PyNET	8.9608	8.2063	8.4138	7.8977	8.0532	7.6866	8.4759	7.9302
PyNET*	9.3716	8.4749	9.2355	8.2926	9.1500	8.2384	9.2523	8.3353
PyNetCA	8.4343	7.8880	7.9795	7.6717	7.8525	7.6036	8.0888	7.7211
InvertISP	7.5380	7.3978	6.7139	6.9881	5.9043	6.2299	6.7187	6.8720
MV-ISPNet	8.1110	7.8820	7.4862	7.2422	7.4597	7.1434	7.6856	7.4225
CameraNet	771.8750	392.2857	771.8755	392.2860	771.8743	392.2864	771.8749	392.2860
AWNet	8.6385	8.3241	8.4038	8.1231	7.0224	7.4135	8.0216	7.9536
Swin-Transformer	7.4143	7.4885	7.2316	7.4287	6.4391	6.7594	7.0284	7.2255
UNet	8.8706	8.2537	8.2972	7.8604	8.2363	7.7737	8.4680	7.9626
UNet*	8.8747	8.2149	8.4267	7.9218	8.4236	8.0117	8.5750	8.0495
eSL	7.8569	7.7214	6.7929	7.0928	6.2029	6.6194	6.9509	7.1445
EV-UNet	8.5472	8.0862	8.1830	7.8198	8.0310	7.7055	8.2537	7.8705



(a) Indoor (b) Outdoor

Figure 15: Event counts in indoor and outdoor scenes. We randomly selected an indoor scene and an outdoor scene. The indoor scene has a strong periodic change, while the outdoor scene does not have a strong periodic change.

F More Discussion

Impact of Indoor Light Flicker: As shown in Fig. 15, indoor light sources exhibit periodic flicker, whereas outdoor light sources do not have this distinct periodic flicker. This issue has also been noted in previous research (Xu et al., 2024a). Addressing this problem is crucial for enhancing image quality. There are two potential solutions: first, applying data augmentation during data input to enable the network to robustly handle flicker issues; and second, using temporal filtering techniques to mitigate the flicker problem.

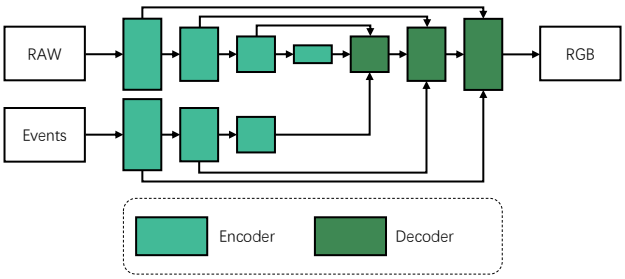


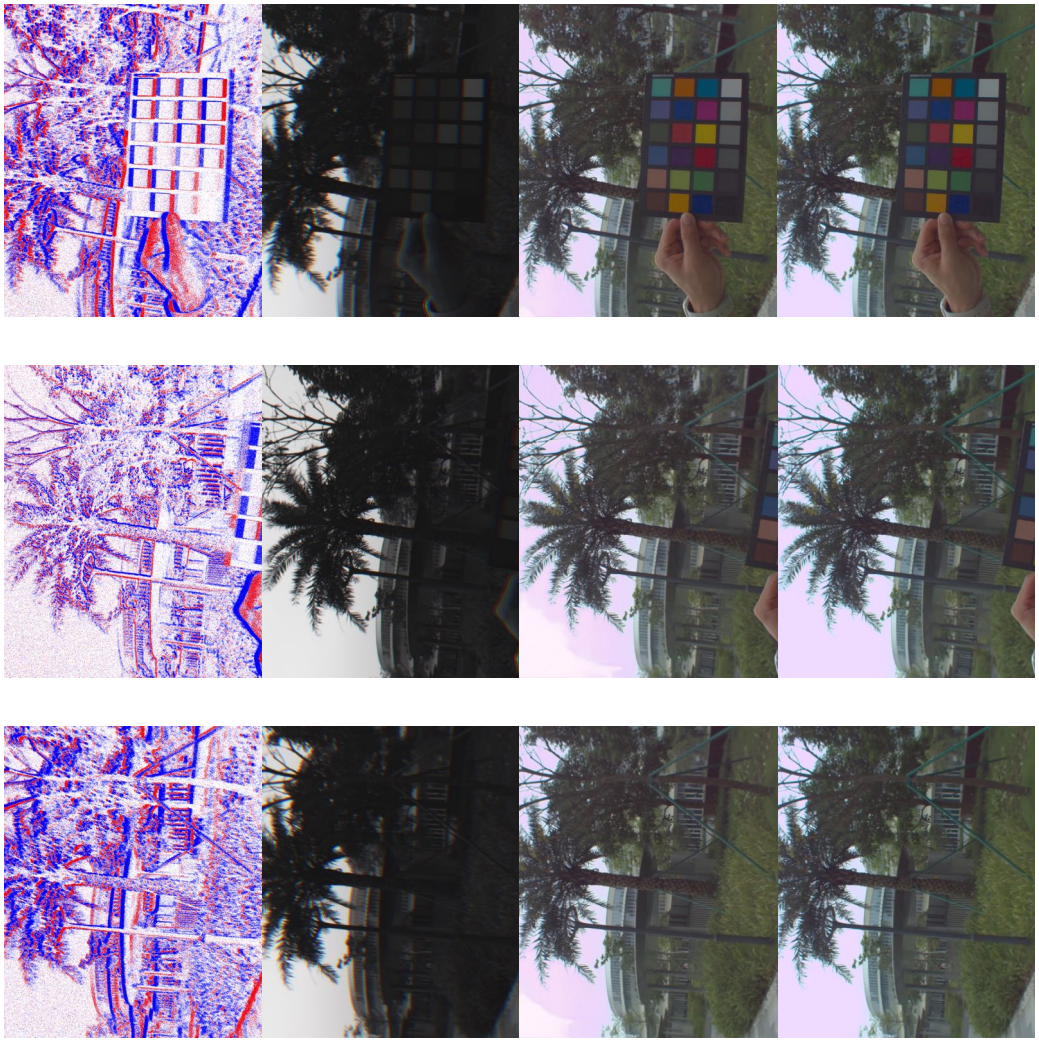
Figure 16: EV-UNet framework.

Table 8: Comparison of Methods on HVS ISP Dataset indoor and outdoor scenes.

	Out-Average			In-Average			All-Average		
Methods	**PSNR**↑	**SSIM**↑	L_1↓	**PSNR**↑	**SSIM**↑	L_1↓	**PSNR**↑	**SSIM**↑	L_1↓
PyNET	32.47	0.9785	0.0180	11.97	0.7664	0.2412	22.22	0.8725	0.1296
PyNET*	29.37	0.9652	0.0265	17.36	0.8437	0.1494	23.37	0.9044	0.0880
PyNetCA	31.76	0.9762	0.0207	27.72	0.9431	0.0540	29.74	0.9596	0.0373
InvertISP	27.59	0.9364	0.0281	28.27	0.9392	0.0254	27.93	0.9378	0.0268
MV-ISPNet	29.76	0.9662	0.0232	31.12	0.9664	0.0207	30.44	0.9663	0.0219
CameraNet	11.36	0.2618	0.2266	13.04	0.2591	0.2013	12.20	0.2605	0.2139
CameraNet*	12.30	0.3953	0.2096	12.68	0.2672	0.2073	12.49	0.3312	0.2085
AWNet	17.04	0.9180	0.0879	27.03	0.9356	0.0567	22.04	0.9268	0.0723
Swin-Transformer	25.24	0.9463	0.0354	25.80	0.9481	0.0304	25.52	0.9472	0.0329
UNet	24.51	0.9634	0.0354	15.70	0.8913	0.1124	20.11	0.9274	0.0739
UNet*	28.17	0.9685	0.0265	31.76	0.9705	0.0188	29.97	0.9695	0.0226
eSL-Net	23.02	0.9294	0.0473	26.13	0.9464	0.0381	24.57	0.9379	0.0427
EV-UNet	30.11	0.9698	0.0225	26.04	0.9388	0.0640	28.08	0.9543	0.0432

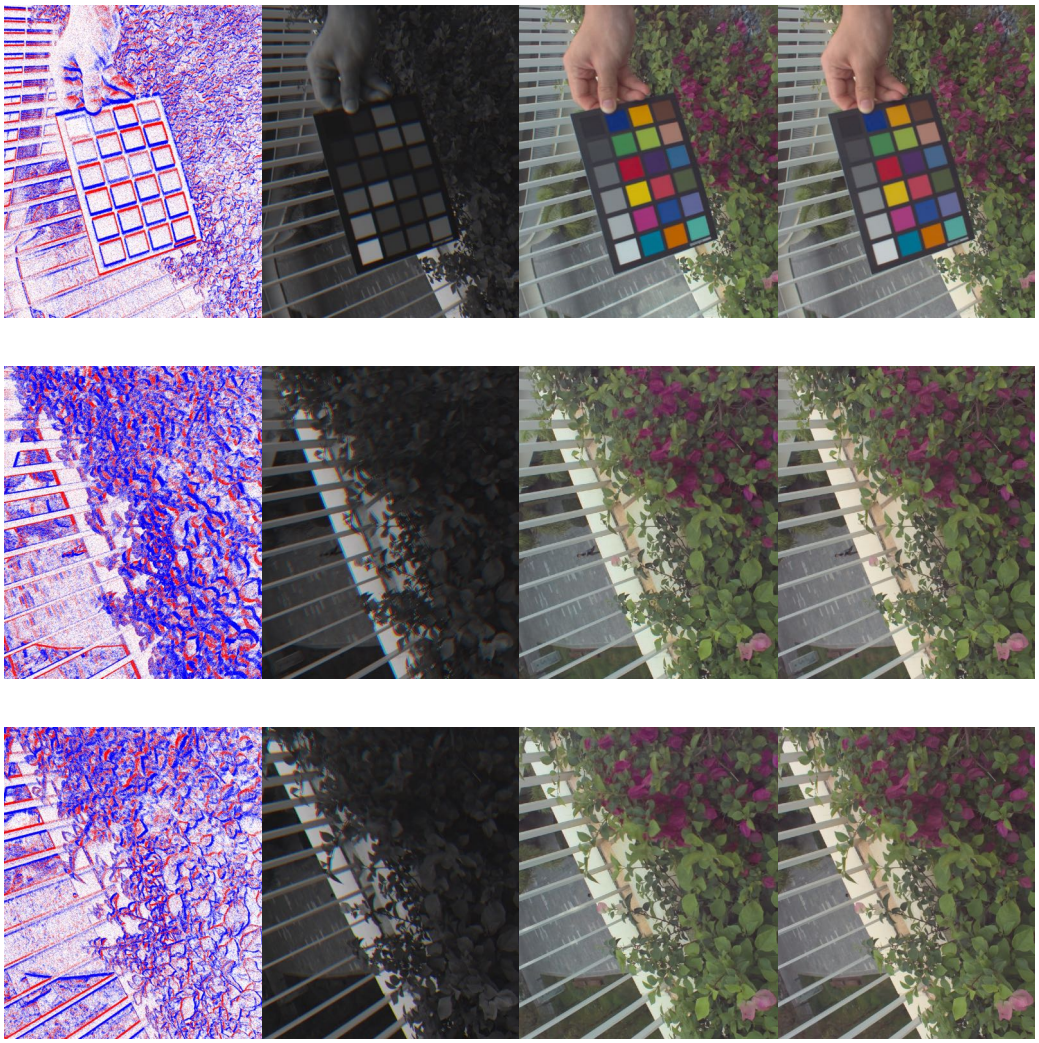
Network Structure of EV-UNet: EV-UNet integrates an event encoding branch into the existing UNet architecture, adding the results of both encoders during the decoding process. Despite this being a simple attempt, we observed that incorporating events can significantly enhance performance in outdoor scenes. For more detailed visual results, please refer to Fig. 17 and Fig. 18.

Analysis of Overall and Scene-Specific Performance: The results in Tab. 8 reveal both overall performance trends and context-specific strengths. For example, UNet demonstrates strong robustness with an `All-Average` PSNR of 29.97, performing well across diverse scenarios. Similarly, MV-ISPNet excels in outdoor scenes, but its performance drops indoors. These findings underline the need to consider scene specific impacts when applying ISP methods, as overall metrics do not always reflect performance in individual contexts. Future research should focus on adapting ISP methods to specific scenarios to ensure optimal outcomes across diverse settings.



(a) Events (b) RAW (c) EV-UNet (d) Good RGB

Figure 17: More visualization results.



(a) Events (b) RAW (c) EV-UNet (d) Good RGB

Figure 18: More visualization results.

Listing 1: MATLAB ISP Code.

```
function ans_colors = RGBE_ISP(
    raw_npz_file,
    json_file,
    has_known_colors,
    known_colors,
    gamma)
    % RGBE_ISP: ISP for RGBE data.
    % raw_npz_file: the raw data file.
    % json_file: the json file for color card.
    % has_known_colors: has known colors.
    % known_colors: the known colors.
    % gamma: gamma value.
    fprintf('RGBE_ISP:\n');
    fprintf('   raw_npz_file :%s\n', raw_npz_file);
    fprintf('   json_file    :%s\n', json_file);
    % Read the raw data for black level calibration.
    pth_blcraw = './rawdata/FixedPatternNoise.npy';
    img_blc = readNPY(pth_blcraw);
    img_blc = double(img_blc) / 242.0;
    % get the file name
    [pathstr, file_name, ext] = fileparts(raw_npz_file);
    % Read rge quad raw data.
    % 242 is the max value of the raw data. The raw data is 8bit.
    % 242 = 255 * 0.95. The 0.95 is the saturation level.
    img_quad = readNPY(raw_npz_file);
    img_quad = max(0, min(img_quad, 242));
    img_quad = double(img_quad) / 242.0;
    img_quad = img_quad - img_blc;
    % clip the value to [0, 1]
    img_quad = max(0, min(img_quad, 1));
    [height, width] = size(img_quad);
    % demosaic the quad raw data.
    % This function can be found in the following link.
    % https://www.mathworks.com/matlabcentral/fileexchange/
    % 116085-quadbayer-cfa-modified-gradient-based-demosaicing
    img_rgb = quad_bayer_demosaic_full(
        img_quad, height, width, 'grgb', 0, 0);
    if has_known_colors
        colors = known_colors;
        ans_colors = [];
    else
        vertex_pts = get_color_card_coords_from_json(json_file);
        % check the vertex_pts has 4 points
        if size(vertex_pts, 1) ~= 4
            disp('Error: vertex_pts has not 4 points');
            return;
        end
        % get the colors from the image given 4 points' location
        % The vertex_pts is the 4 points of the color card.
        [colors, coord] = checker2colors(
            img_rgb, [4, 6], 'mode', 'auto',
            'show', false, 'vertex_pts', vertex_pts);
        % save colors to file
        color_file = sprintf('%s/%s_colors.mat', pathstr, file_name);
        save(color_file, 'colors');
        % the colors will be reture value.
        ans_colors = colors
        % check NaN value in colors. if has NaN value, return.
        if any(isnan(colors))
            disp('Error: colors has NaN value');
            fprintf('raw_npz_file: %s\n', raw_npz_file);
            return;
        end
```

```
    end
    % white balance
    wb_multipliers = [
        colors(21, 2) / colors(21, 1),
        1.0,
        colors(21, 2) / colors(21, 3)];
    img_wb = img_rgb;
    img_wb(:, :, 1) = img_wb(:, :, 1) * wb_multipliers(1);
    img_wb(:, :, 3) = img_wb(:, :, 3) * wb_multipliers(3);
    img_wb = max(0, min(img_wb, 1));
    % denoise using rgb BM3D with default parameter
    randn('seed', 0);
    sigma = 25;
    [~, img_denoise] = CBM3D(1, img_wb, sigma);
    % color correction.
    % The color card colors are sRGB from
    % the document of the color card,
    % treated as groundtruth sRGB under D65
    srgb = [
            112, 76, 60;
            197, 145, 125;
            87, 120, 155;
            82, 106, 60;
            126, 125, 174;
            98, 187, 166;
            238, 158, 25;
            157, 188, 54;
            83, 58, 106;
            195, 79, 95;
            58, 88, 159;
            222, 118, 32;
            25, 55, 135;
            57, 146, 64;
            186, 26, 51;
            245, 205, 0;
            192, 75, 145;
            0, 127, 159;
            43, 41, 43;
            80, 80, 78;
            122, 118, 116;
            161, 157, 154;
            202, 198, 195;
            249, 242, 238;
            ];
    srgb = srgb / 255.0; %normlization
    srgb = srgb .^ 2.2;  %sRGB to linear sRGB
    colors_wb = colors .* wb_multipliers % white balance correction
    % compute the color correction matrix.
    [cam2xyz, scale, ~, ~] = ccmtrain(colors_wb, ...
        srgb, 'omitlightness', true, 'preservewhite', true, ...
        'model', 'linear3x3', 'targetcolorspace', 'sRGB', ...
        'whitepoint', whitepoint('d65'));
    % apply the color correction matrix.
    lin_srgb = apply_cmatrix(
        img_denoise * (scale * 0.9),transpose(cam2xyz));
    lin_srgb = max(0, min(lin_srgb, 1));
    % gamma correction.
    img_srgb = lin_srgb .^ gamma;
    img_srgb = max(0, min(img_srgb, 1));
    good_rgb_file = sprintf(
        '%s/%s_good_rgb.png', pathstr, file_name);
    imwrite(img_srgb, good_rgb_file);
    fprintf('DONE: %s', good_rgb_file);
end
```