Dynamic Exposure Burst Image Restoration
— *Supplementary Material* —

Woohyeok Kim Jaesung Rim Daeyeon Kim Sunghyun Cho

POSTECH

In this supplementary material, we first present additional details on BAENet (Sec. S1), followed by descriptions of the noise synthesis process (Sec. S2), RAW conversion pipeline (Sec. S3), and details of the dataset (Sec. S4). We then describe the camera system for real-world evaluation (Sec. S5) and computational cost analysis (Sec. S6). Next, we present additional experimental results (Sec. S7). Finally, we include additional qualitative results (Sec. S8).

S1. Implementation Details of BAENet

Normalization of Inputs BAENet takes a preview image along with gain g_p and motion magnitude m_p as inputs. Since g_p and m_p have different scales with large variations, we normalize them to the range $[0, 1]$ as follows:

$$g_p = (\hat{g}_p - g_{\text{min}})/(g_{\text{max}} - g_{\text{min}}) \in [0, 1], \tag{S1}$$

$$m_p = \min\left(\hat{m}_p/m_{\text{thr}}, 1\right) \in [0, 1], \tag{S2}$$

where $\hat{g}_p$ and $\hat{m}_p$ are the gain and motion magnitude before the normalization, respectively. g_{min} and g_{max} represent the minimum and maximum gain values for the shooting environment, respectively. We set $g_{\text{min}} = 51200$ and $g_{\text{max}} = 102400$ to account for extremely low-light conditions. m_{thr} denotes the threshold of motion magnitude and is set to 20. We empirically found that truncating motion values beyond this threshold improves performance.

Network Architecture For BAENet, we adopt MobileNetV2 [11], and construct the input by concatenating g_p and m_p to each pixel position of $I_p \in \mathbb{R}^{H\times W\times 3}$, then feed the concatenated tensor $B_{\text{input}} \in \mathbb{R}^{H\times W\times 5}$ into BAENet. We modify the input channels of the first convolution layer of MobileNetV2 to 5 to match the input of BAENet, and change the output channels of the last fully connected layer to $n + 1$ in order to predict the exposure times for n burst images. Here, the $(n + 1)$-th element of the output vector allows the sum of the n exposure times to be shorter than the upper bound, if necessary.

S2. Details of Noise Synthesis

In the differentiable burst simulator, we synthesize realistic noise using a heteroscedastic Gaussian distribution, following [17]:

$$N \sim \mathcal{N}\left(0, \lambda_{\text{read}} + \lambda_{\text{shot}} S_{s,e}\right), \tag{S3}$$

where λ_{shot} and λ_{read} represent the shot noise and read noise parameters, respectively, and $S_{s,e}$ denotes the integration of scene radiance during the exposure duration, corresponding to Eq. (7) in the main paper. The amount of noise should be proportional to the gain of each burst image g_i. In the following section, we describe how to determine the noise parameters according to g_i. In this paper, we use the term "gain" to refer to the linear gain, which is equivalent to the camera ISO.

Given a preview image, we compute the gain of each burst image g_i as:

$$g_i = g_p \cdot \frac{t_p}{t_i}, \tag{S4}$$

where t_i is the exposure time estimated by BAENet for each burst image, and g_p and t_p are the gain and exposure time of the preview image, respectively. To simulate extremely low-light images, we randomly sample g_p from a uniform distribution within the range $[51200, 102400]$.

Based on g_i, we compute the shot noise parameter λ^i_{shot} and the read noise parameter λ^i_{read} for each burst image. To this end, we first calibrated the shot noise and read noise parameters of our camera system across a range of gains, from 100 to 12,800. Then, we model the relationship between gains and the shot and read noise parameters, following [3]. The shot noise parameter is linearly proportional to the gain. For each burst image, we model the shot noise parameter λ^i_{shot} as:

$$\lambda^i_{shot} = 9.2857\text{e}{-07} \times g_i + 8.1006\text{e}{-05}, \tag{S5}$$

The coefficients of Eq. (S5) are estimated from the calibrated shot noise parameters in our camera system. Additionally, read noise parameters are known to be linearly proportional to the shot noise parameters in the log domain. We

model the read noise parameter λ^i_{read} as:

$$\log(\lambda^i_{read}) = 2.2282 \times \log(\lambda^i_{shot}) + 0.45982, \quad \text{(S6)}$$

The coefficients of Eq. (S6) are also estimated from the calibrated read noise and shot noise parameters. For each burst image, we compute λ^i_{shot} and λ^i_{read} according to g_i and then synthesize noisy burst images using Eq. (S3).

S3. Details of RAW conversion

In our dataset, we convert sRGB video clips into RAW video clips. We first apply gamma expansion to convert the video clip into the linear sRGB space, then convert its color space to the RAW color space using color correction matrices (CCMs). Next, we apply inverse white balance and mosaic the resulting videos to the RGGB Bayer pattern. In this section, we describe the details of color space conversion and inverse white balance.

Conversion to RAW Color Space Camera systems capture images in the RAW color space and convert them to sRGB images using color correction matrices (CCMs). Using the inverse of CCMs, we can convert the sRGB videos in our dataset to the RAW color space as follows:

$$\begin{bmatrix} R_{\text{RAW}} \\ G_{\text{RAW}} \\ B_{\text{RAW}} \end{bmatrix} = \text{CCM}^{-1} \cdot \begin{bmatrix} R_{\text{sRGB}} \\ G_{\text{sRGB}} \\ B_{\text{sRGB}} \end{bmatrix}, \quad \text{(S7)}$$

where $(R_{\text{sRGB}}, G_{\text{sRGB}}, B_{\text{sRGB}})$ is an RGB color of the sRGB color space. $(R_{\text{RAW}}, G_{\text{RAW}}, B_{\text{RAW}})$ is the corresponding RGB color converted to the RAW color space. CCM is a color correction matrix. To obtain the matrix, we calibrated three CCMs using the target camera (Basler a2A1920-160ucBAS) under three different scenes. We randomly sample CCM from the three estimated matrices for each video clip and convert the video clip to the RAW color space. This reflects the fact that the CCM of real camera systems can vary under different light sources.

Inverse White Balance For each video clip, we randomly sample white balance gains to reflect varying white balance adjustments in real-world scenarios. Then, we perform inverse white balance as follows:

$$R = R' \cdot \frac{g_{RGB}}{g_R}, \quad G = G' \cdot g_{RGB}, \quad B = B' \cdot \frac{g_{RGB}}{g_B}, \quad \text{(S8)}$$

where R', G', B' and R, G, B are the pixel values of the RGB channels before and after inverse white balance, respectively. g_{RGB} represents the total gain for all RGB channels, while g_R and g_B are the individual white balance gains for red and blue channels. The white balance gain for the green channel is typically set to one, so we omit it in Eq. (S8). We randomly sample g_{RGB} from a Gaussian distribution $\mathcal{N}(0.8, 0.1)$. g_R and g_B are sampled from uniform distributions $\mathcal{U}(1.9, 2.4)$ and $\mathcal{U}(1.5, 1.9)$, respectively.

S4. Details of the Dataset

Along with scene radiance sequences with camera and object motion from the GoPro dataset [7], we include static sequences without such motion. To achieve this, we utilize ground-truth sharp images from the RealBlur dataset [9]. Specifically, we sample a sharp sRGB image from the RealBlur dataset and convert it into the camera RAW color space as described in Sec. 5.1 of the main paper. We then generate a scene radiance sequence by duplicating it. Although assuming no camera or object motion in burst shots is less realistic, we found empirically that including static scenes stabilizes the training process and enhances the performance of our framework. Finally, we generated $\mathcal{D}_{\text{restore}}$ and $\mathcal{D}_{\text{BAENet}}$ containing a total of 4,092 and 1,127 sequences, respectively, with 3,728 and 1,036 from the GoPro dataset and 364 and 91 from the RealBlur dataset. For evaluation of our method, we also generated a test set consisting of 532 scene-radiance sequences from the GoPro dataset.

Since our synthetic training dataset uses frames from daytime high-FPS videos [7] as the source images for burst simulation, a potential sim-to-real gap may arise. As shown in Fig. 2 of the main paper, the preview and input burst images are captured in daytime conditions, which differ from the extremely low-light scenarios targeted by our method. Nevertheless, prior work [10] has shown that models trained on blurred images synthesized from daytime high-FPS videos with added noise can generalize well to real low-light data [9]. Consistent with these findings, our real-capture experiments presented in Sec. 6.2 of the main paper demonstrate that the proposed method performs reliably in real-world low-light scenarios. At the same time, the sim-to-real gap cannot be entirely eliminated and remains a natural limitation when training on synthetic data. Addressing this gap is an important direction for future work.

S5. Camera System for Real-world Evaluation

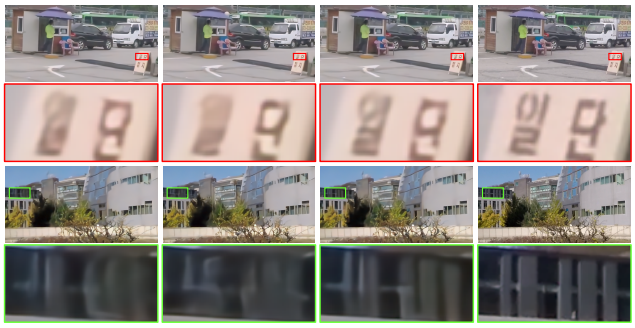
In the real-world capturing setup described in Sec. 6.2 of the main paper, our handheld dual-camera system consists of two Basler a2A1920-160ucBAS cameras, each equipped with an Edmund 6 mm C-Series lens. One camera captures burst images with exposure times predicted by BAENet, while the other uses a predefined exposure setting of $\{8, 24, 40, 56\}/1920$ seconds. The system was connected to a Samsung Galaxy Book4 Ultra NT960XGP laptop, and we captured 142 burst images in low-light environments.

S6. Computational Cost

The computational workload of our pipeline consists of three components: motion estimation from preview images using RAFT-small [13], BAENet, and burst image restoration. Other methods in Tab. 1 rely on heuristic strategies or lightweight predictors for exposure selection, so their

Components	Motion estimation	BAENet	Burst image restoration
FLOPs (G)	65	6	1769

Table S1. Computational cost of each module in our pipeline.



{8, 24, 40, 56} {8, 16, 32, 64} DEBIR (Ours) Ground truth

Figure S1. Qualitative comparison with another predefined-exposure schedule on our test set.

Metric	$\{8, 24, 40, 56\}$	$\{8, 16, 32, 64\}$	DEBIR (Ours)
PSNR↑	35.04	35.07	**35.32**
SSIM↑	0.9481	0.9482	**0.9519**
LPIPS↓	0.164	0.163	**0.154**

Table S2. Quantitative comparison with another predefined-exposure schedule on our test set. The best and second-best results are in bold and blue, respectively.

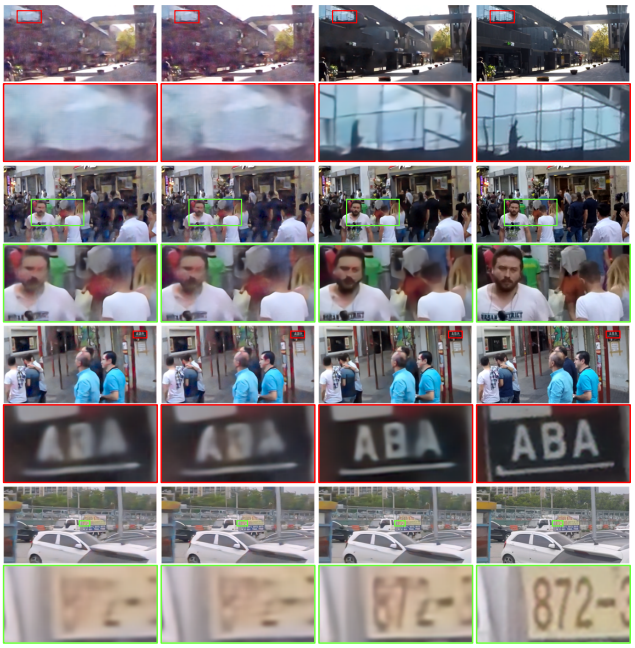
computational cost mainly arises from burst image restoration (1769G FLOPs). In comparison, DEBIR additionally requires motion estimation and exposure prediction using BAENet, which together introduce only 71G FLOPs as reported in Tab. S1. Therefore, the additional overhead of DEBIR is negligible compared to the overall cost of burst image restoration.

S7. Additional Experiments

Predefined Exposure Schedule In the main paper, we compared against a predefined-exposure baseline [6, 17] based on an arithmetic schedule, $\{8, 24, 40, 56\}/1920$ seconds, where the exposure times increase by a fixed step of $16/1920$ seconds. Since HDR pipelines often employ exponentially increasing exposure times, we additionally evaluated a geometric schedule, $\{8, 16, 32, 64\}/1920$ seconds, which follows a constant ratio of 2. As shown in Fig. S1, DEBIR produces finer details than the predefined schedules. Consistently, Tab. S2 shows that both predefined schedules yield similar but overall inferior performance, whereas our DEBIR model achieves the best results across all metrics. This is because predefined exposure bracketing follows a fixed schedule that cannot account for varying shooting environments, whereas DEBIR can adapt to the scene and thus achieves better results.

Performance Gap across Noise Levels To further analyze the results in Tab. 1 of the main paper, we evaluate performance across different noise levels. Our test set is synthesized with noise corresponding to ISO values sampled from the range [51200, 102400]. We divide the test set into five bins with evenly spaced ISO intervals and evaluate performance in each bin, as summarized in Tab. S3.

The first two rows in Tab. S3 correspond to the predefined-exposure baseline Exposure Bracket [6, 17] using arithmetic and geometric exposure schedules $\{8, 24, 40, 56\}/1920$ and $\{8, 16, 32, 64\}/1920$ seconds, respectively. Across all noise levels (columns 1–5 in Tab. S3, with increasing ISO values), DEBIR consistently achieves higher PSNR than Exposure Bracket, and the performance gap becomes larger as the ISO value increases. We further construct a new test set using the same scenes with ISO values sampled from a higher range [153600, 204800], resulting in significantly stronger noise. The corresponding result is reported in column 6 (New) of Tab. S3, where the performance gap between Exposure Bracket and DEBIR increases to 0.83 dB.



{8, 24, 40, 56} {8, 16, 32, 64} DEBIR (Ours) Ground truth

Figure S2. Qualitative comparison with Exposure Bracket [6, 17] on the new test set with stronger noise.

Methods	Original (Tab. 1)					New
	1	2	3	4	5	
$\{8, 24, 40, 56\}$	35.75	35.40	35.15	34.35	34.70	32.04
$\{8, 16, 32, 64\}$	35.81	35.43	35.20	34.36	34.71	32.00
DEBIR (Ours)	**35.97**	**35.64**	**35.43**	**34.65**	**35.03**	**32.87**

Table S3. PSNR comparison with Exposure Bracket. Columns 1–5 correspond to ISO-based subsets of our test set, while column 6 corresponds to a new test set with stronger noise. The best and second-best results are shown in bold and blue, respectively.

Test set	Dig. Gim. [4]	Act. S-L [16]	Avg. AE [2]	Grad. AE [12]	Exp. Brk. [6, 17]	DEBIR (Ours)
REDS + Basler	32.53	32.44	33.44	33.68	33.60	**33.86**
REDS + Pixel	31.44	31.26	32.29	32.44	32.45	**32.50**

Table S4. Quantitative comparison of PSNR on new test sets. The best and second-best results are in bold and blue, respectively.

i-th burst image	w/o I_1	w/o I_2	w/o I_3	w/o I_4	Full
PSNR↑	13.71	32.12	33.91	34.89	**35.32**
SSIM↑	0.1059	0.9161	0.9367	0.9461	**0.9519**
LPIPS↓	0.627	0.198	0.183	0.167	**0.154**

Table S5. Analysis of the impact of each burst image on performance. Here, I_i refers to the i-th burst image. The best and second-best results are in bold and blue, respectively.

Qualitative results in Fig. S2 also support this observation. DEBIR better preserves fine details while producing fewer color artifacts caused by denoising failures. Overall, these results demonstrate that DEBIR shows a clear advantage under harsher low-light conditions with stronger noise. By adaptively predicting exposure times according to the shooting environment, DEBIR enables more effective burst image restoration.

Generalization Capability In image restoration tasks, neural networks trained on data from a specific dataset or camera are not necessarily expected to generalize well to different datasets or devices [9]. Our setup follows common practice in image restoration, where models are typically trained for a target camera using datasets tailored to that device. However, it may still be questioned whether our model achieves strong performance only on the datasets or the noise characteristics of the target camera used in our experiments, i.e., whether the reported performance is simply a result of overfitting to the datasets used. To this end, we conducted an additional study on generalizability.

Specifically, we generated two test sets using the REDS dataset [8] with noise parameters from two different cameras: a Basler camera, which was used in our main experiments, and a Google Pixel. For the Google Pixel, we used the calibrated noise parameters provided in the SIDD dataset [1]. As our training set is generated using sharp frames from the GoPro [7] and RealBlur [9] datasets and the noise characteristics of a Basler camera, these test sets allow us to examine generalization to unseen scenes captured by both the same and a different device.

As shown in Tab. S4, our method outperforms other approaches on 'REDS + Basler', demonstrating strong generalization to novel scenes from the same camera. Furthermore, while cross-device generalization is inherently challenging, our results on 'REDS + Pixel' suggest that the model may still outperform competing methods even on images captured by a different camera. Furthermore, since our pipeline synthesizes training data from camera parameters, adapting the framework to a different target camera is straightforward in practice.

The Impact of Each Burst Image In this study, we examine which image among the burst images is most important in our pipeline. To verify this, we replace each burst image with a zero-filled image and evaluate the performance. Tab. S5 shows that replacing any burst image significantly reduces performance, indicating that all burst images play a crucial role in restoration. Notably, the first burst image, which is the base frame in the burst image restoration network, has the most significant impact on performance. The influence gradually decreases for subsequent frames.

Comparison with RL-based Methods Several RL-based methods [14, 15] have been proposed for automatic exposure control. However, most of them target HDR reconstruction and are not well suited for burst image restoration. AdaptiveAE [15] predicts exposure times in a stepwise manner, resulting in a sequential decision-and-capture pipeline that is inefficient and unsuitable for burst image restoration scenarios. Another RL-based method [14] assumes static scenes and operates on a small, fixed number of exposures (e.g., three), which makes it difficult to extend methodologically to burst image restoration settings that typically require more frames.

Nevertheless, we evaluate a publicly available RL-based method [14] for comparison. To ensure a fair comparison, we fine-tuned Burstormer [5] using exposure times predicted by the policy network of [14]. The corresponding burst images were synthesized using our burst simulator. Using this approach, the method achieved 34.71 dB PSNR, which is 0.61 dB lower than DEBIR. While our focus is burst image restoration, extending DEBIR to support HDR reconstruction is an interesting direction for future work.

Motion Estimation Reliability In our implementation, preview images are captured with an exposure time of $1/120$ seconds, which we empirically found to provide sufficiently sharp previews for reliable flow estimation, whereas longer exposure times introduce motion blur. Under this setting, we further evaluated the robustness of motion estimation by estimating the motion magnitude m_p using noise-free preview images and observed only a negligible performance difference of 0.04 dB. This robustness stems from how the motion cue m_p is defined in our pipeline: optical flow is computed on downsampled preview images and aggregated into a single scalar by averaging over the entire frame, intentionally capturing only coarse motion information. Although optical flow estimation may degrade under more severe conditions, the estimator itself can be further fine-tuned to improve robustness if necessary. Moreover, our framework is not restricted to optical flow. Alternative motion cues such as gyroscope signals can also be incorporated without modifying the pipeline.

S8. Additional Results

We present additional qualitative results on our test set (Fig. S3) and real-world results obtained with our camera system (Fig. S4). These results demonstrate that DEBIR yields visually superior outcomes over other methods, confirming its effectiveness.

References

[1] 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 (CVPR)*, 2018. 4

[2] ARM. Mali-C71. https://www.arm.com/products/silicon-ip-multimedia/image-signal-processor/mali-c71ae, 2020. Camera product. 4

[3] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[4] Omer Dahary, Matan Jacoby, and Alex M Bronstein. Digital gimbal: End-to-end deep image stabilization with learnable exposure times. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4

[5] Akshay Dudhane, Syed Waqas Zamir, Salman Khan, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Burstormer: Burst image restoration and enhancement transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 4

[6] Sanghyun Kim, Minjung Lee, Woohyeok Kim, Deunsol Jung, Jaesung Rim, Sunghyun Cho, and Minsu Cho. Burst image super-resolution with base frame selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024. 3, 4

[7] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 4

[8] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 4

[9] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2, 4

[10] Jaesung Rim, Geonung Kim, Jungeon Kim, Junyong Lee, Seungyong Lee, and Sunghyun Cho. Realistic blur synthesis for learning image deblurring. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[11] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[12] Inwook Shim, Tae-Hyun Oh, Joon-Young Lee, Jinwook Choi, Dong-Geol Choi, and In So Kweon. Gradient-based camera exposure control for outdoor mobile platforms. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 29(6):1569–1583, 2018. 4

[13] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[14] Zhouxia Wang, Jiawei Zhang, Mude Lin, Jiong Wang, Ping Luo, and Jimmy Ren. Learning a reinforced agent for flexible exposure bracketing selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 4

[15] Tianyi Xu, Fan Zhang, Boxin Shi, Tianfan Xue, and Yujin Wang. Adaptiveae: An adaptive exposure strategy for hdr capturing in dynamic scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2025. 4

[16] Dan Yang, Samu Koskinen, and Joni-Kristian Kämäräinen. Active short-long exposure deblurring. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2022. 4

[17] Zhilu Zhang, Shuohao Zhang, Renlong Wu, Zifei Yan, and Wangmeng Zuo. Exposure bracketing is all you need for a high-quality image. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 1, 3, 4

Figure S3. Additional qualitative results on our test set.

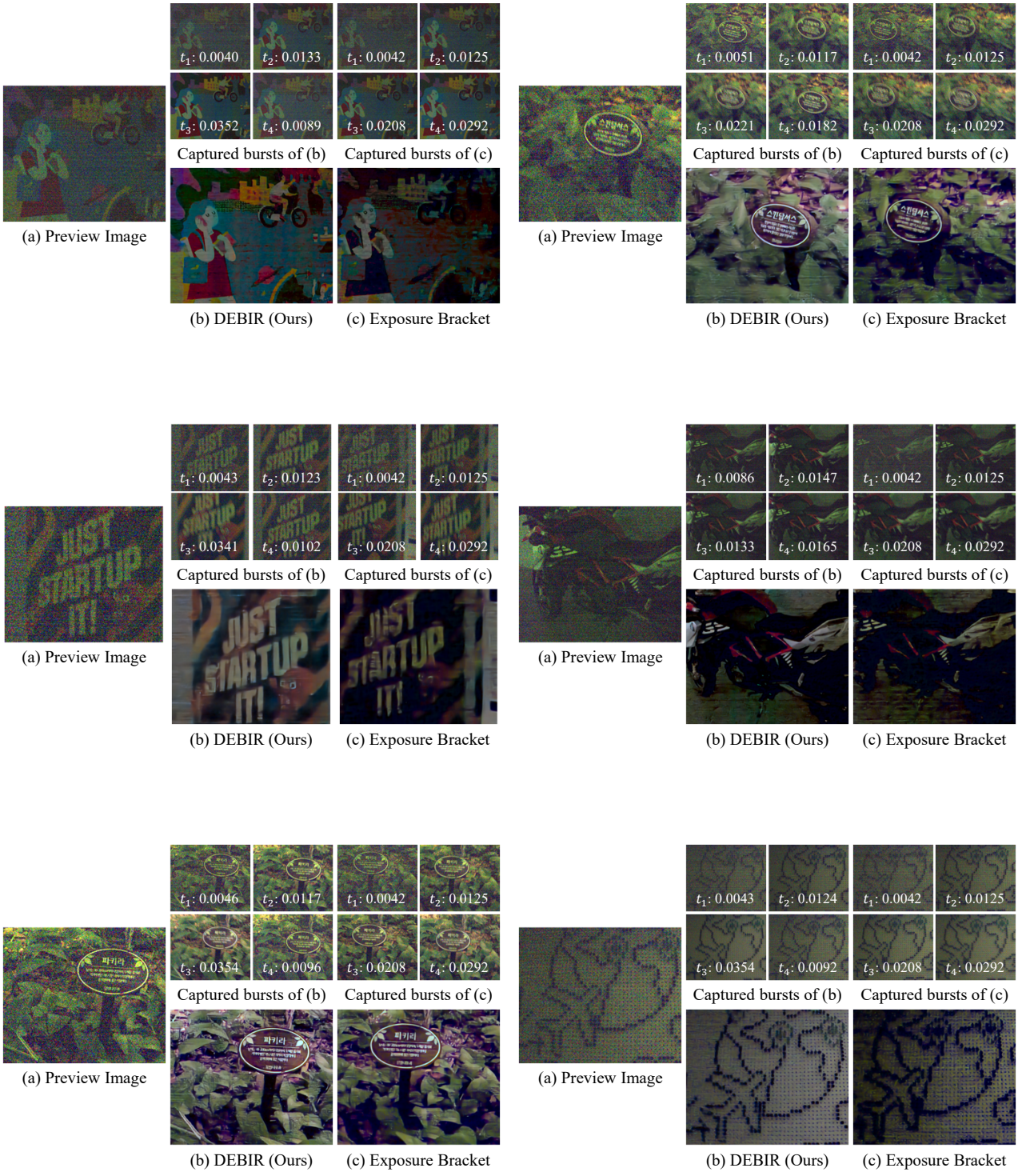


Figure S4. Additional qualitative results using a real-world camera system.