Supervised Raw Video Denoising with a Benchmark Dataset on Dynamic Scenes Supplementary Material

Huanjing Yue Cong Cao Lei Liao Ronghe Chu Jingyu Yang
School of Electrical and Information Engineering, Tianjin University, Tianjin, China
{huanjing.yue, caocong_123, leolei, chu_rh, yjy}@tju.edu.cn

This supplementary material provides details which were not presented in the main paper due to space limitations. In the following, we first present the details for our constructed dataset, including indoor and outdoor datasets. Then, we give the noise synthesis details. Hereafter, we give the comparison results with SMD [2], including the retrained SMD results for our dataset. Finally, we give the user study results on our outdoor dataset. We also give a video demo for the outdoor video denoising results. The readers are encouraged to watch the video for better observation and comparison.

1. Raw Video Dataset

1.1. Captured Raw Video Dataset

As stated in the main paper, we totally captured 11 indoor scenes under 5 different ISO levels ranging from 1600 to 25600. For each static moment, we continuously captured M noisy frames. The averaging of the M frames is the expected noise free frame. We note that there is still slight noise after averaging noisy frames, and we further applied BM3D [4] to the averaged frame to get a totally clean ground truth. Table 1 lists the detailed information for our captured dataset.

Fig. 1 presents 11 indoor scenes in our dataset. Each scene is captured under five different ISO values. The first six scenes are used for training and the last five scenes are used for testing.

We also captured 50 videos for ten outdoor scenes to test the effectiveness of our model trained with indoor scenes. Each video contains 50 frames. Fig. 2 presents the ten outdoor scenes.

1.2. Noise Calibration for a Given Camera

As demonstrated in the main paper, the noise model for raw images is

$$x_p \sim \sigma_s^2 \mathcal{P}(y_p/\sigma_s^2) + \mathcal{N}(0, \sigma_r^2) \tag{1}$$

where x_p is the noisy observation, y_p is the true intensity at pixel p. σ_r and σ_s are parameters for read and shot noise, which vary across images as sensor gain (ISO) changes. We calibrate the noise parameters for given camera by capturing flat-field frames and bias frames. Since σ_s and σ_r are different for different ISO settings, in the following we give details for the parameter estimation for one ISO setting. The estimation process for other ISO settings is similar.

Table 1. Detailed settings for our captured indoor dataset. M is the number of frames used to generate the clean frame. σ is the parameter for BM3D denoising. Each video contains 7 consecutive frames and we capture 11 videos for each ISO setting.

ISO	M	σ	Number of Videos
1600	150	0.125	11
3200	150	0.25	11
6400	250	0.5	11
12800	250	1	11
25600	500	2	11

We put a white paper on a uniformly lit wall to capture flat-field frames. We continuously take two images of the white paper for a specific exposure time, which is denoted by

$$\begin{aligned} x_p^a &= y_p + n_r^a(p) + n_s^a(p) y_p, \\ x_p^b &= y_p + n_r^b(p) + n_s^b(p) y_p, \end{aligned} \tag{2}$$

where x_p^a and x_p^b are the two captured noisy images at position p, $n_r(p)$ represents the read noise and $n_s(p)y_p$ is the shot noise. To avoid the influence of vignetting, we crop a 400×400 patch from the center of the captured image. Since y_p is supposed to be the same for the cropped patch and the noise is random, we utilize the median of the averaging pixels $0.5(x_p^a + x_p^b)$ as the true intensity, i.e. y. The difference between the two images is the noise signal, which is

$$x_p^a - x_p^b = n_r^a(p) - n_r^b(p) + (n_s^a(p) - n_s^b(p)) y_p. \tag{3}$$

The variance of $x^a - x^b$, denoted by σ_d^2, is 2 times of the original variance of x^a. Then we obtain a point $(y_1, \sigma_{d_1}^2)$

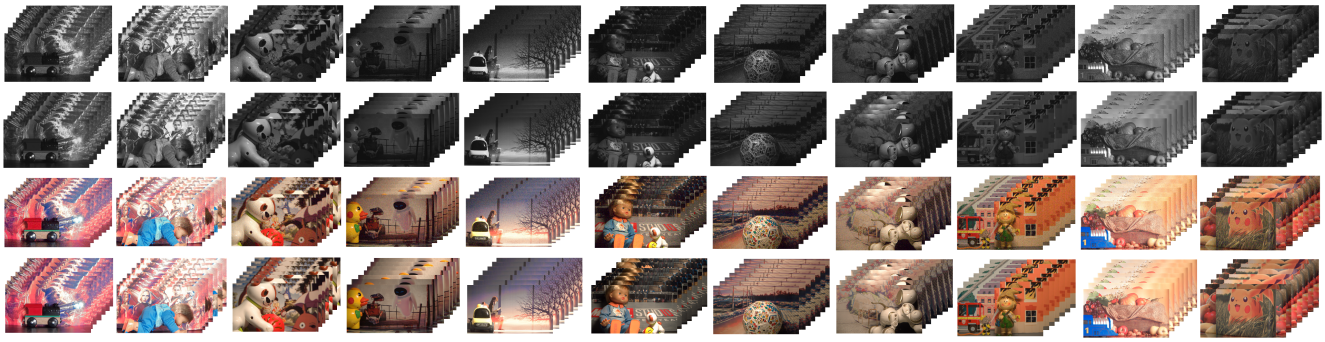

Figure 1. The eleven indoor scenes in our dataset. From top to down, each row lists the raw noisy videos (captured under ISO 25600), raw clean videos, sRGB noisy videos, and sRGB clean videos, respectively. The color videos are generated from raw video using our pre-trained ISP module.

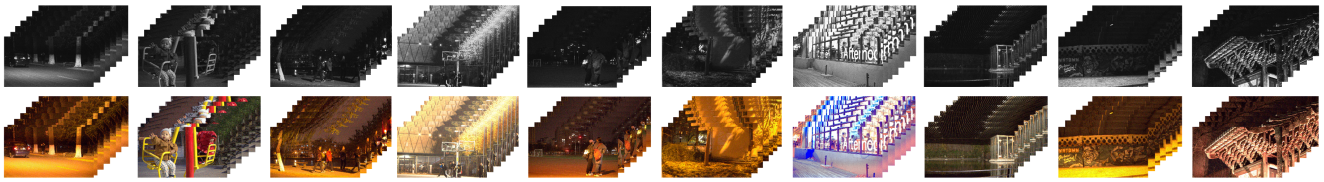

Figure 2. The ten outdoor scenes in our dataset. The top row is the raw noisy videos and the bottom row is the corresponding sRGB noisy videos generated with our pre-trained ISP module.

for current exposure time. Hereafter, we capture the white paper using another exposure time and repeat the above process. Then will get another point $(y_2, \sigma_{d_2}^2)$. We repeat this process for several times and get several points. Then we plot these points and the slope (denoted by k) of the line is the estimated variance $2\sigma_s^2$. As shown in Fig. 3, these points (denoted by blue points) generally form a straight line except for the points near the clipping threshold. σ_s^2 can be derived by $\frac{k}{2}$. Fig. 3 presents the lines for the five ISO settings and the corresponding estimated σ_s^2 for noise synthesis.

σ_r is estimated by capturing bias frames. We cover the lens with the camera cover and then we capture the bias frame in a dark room. For each ISO setting, we capture five bias frames. Since there is no shot noise in bias frames, the variance of the bias frame is caused by read noise. We utilize the average of the five estimated variances as the final σ_r^2 for a specific ISO.

2. Quantitative Comparison Results

In the main paper, we only present the average denoising results for all the ISO settings. Here, we further present the average denoising results for each ISO setting in Table 2.

We give three results for SMD. The first result, denoted by SMD, is generated with their released model and our raw video is preprocessed with their settings. In order to compare with our method in the full-resolution result, we did not utilize the binning process in SMD and utilize widely used demosaicing process used in [1] to preprocess our dataset for SMD. The second result, denoted by SMD*, is generated by retraining SMD with our dataset and the VBM4D preprocessing used in the original SMD code is removed. The third result, denoted by SMD-R is generated by retraining SMD with our dataset and the VBM4D preprocessing is included. TOFlow, EDVR, and DIDN are retrained with our dataset.

It can be observed that our method greatly outperforms the denoising methods conducted on sRGB domain. Note that although SMD-R contains VBM4D preprocessing, it still cannot remove the noise well [1].

3. Qualitative Comparison Results

3.1. Comparison with SMD

Since the SMD dataset is constructed by static scenes, we directly process the SMD dataset without retraining. Fig. 4 presents the visual comparison results for two outdoor scenes. SMD prefers the results with large digital gains. However, this tends to over-expose the bright regions in images. In contrast, we prefer to brighten the images to a moderate level and remove the noise. When SMD and our method utilize the same digital gain, our result is more natural than SMD. There is no over exposing in our result and the noise is removed clearly. Note that, we can also deal with larger digital gains if we retrain our model with noisy-clean pairs where larger digital gains are applied for

[1]The results of SMD-R may be improved if we tune the parameters for VBM4D. However it is time consuming and tuning VBM4D cannot bring more than 1 dB gain.

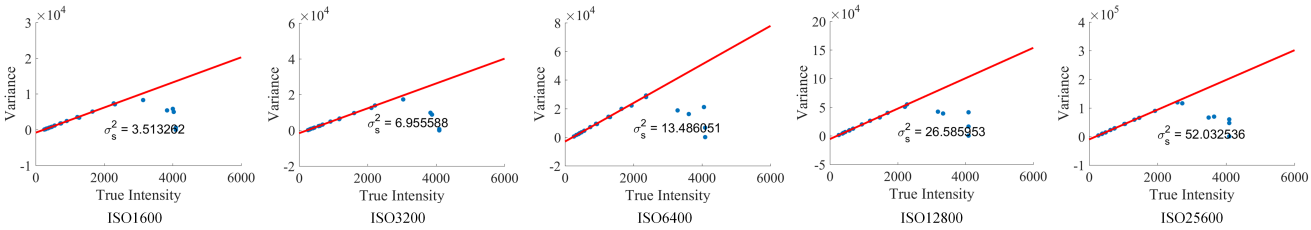

Figure 3. The estimated σ_s for five ISO settings.

Table 2. Comparison with state-of-the-art denoising methods. Each row lists the average denoising results in raw (or sRGB) domain for the scenes captured under the specified ISO and the last row is the average denoising results for all the ISO settings. The results of Ours$^-$ are obtained by training the proposed model with only synthetic noisy videos. The best results are highlighted in bold and the second best results are underlined.

ISO			Noisy	ViDeNN [3]	VBM4D [5]	TOFlow [7]	SMD [2]	SMD*	SMD-R	EDVR [6]	DIDN [8]	Ours$^-$	Ours
1600	Raw	PSNR	38.57	-	-	-	-	-	-	-	47.00	47.14	**47.74**
		SSIM	0.921	-	-	-	-	-	-	-	0.993	0.993	**0.994**
	sRGB	PSNR	37.41	35.44	39.34	37.61	26.59	37.81	36.30	42.10	41.85	42.24	**43.13**
		SSIM	0.922	0.966	0.967	0.964	0.923	0.969	0.968	0.984	0.985	0.985	**0.988**
3200	Raw	PSNR	35.16	-	-	-	-	-	-	-	45.02	45.26	**45.91**
		SSIM	0.854	-	-	-	-	-	-	-	0.989	0.990	**0.991**
	sRGB	PSNR	34.90	34.37	36.62	36.97	26.51	37.07	36.36	41.03	40.65	41.13	**41.99**
		SSIM	0.871	0.946	0.951	0.958	0.918	0.964	0.965	0.980	0.980	0.982	**0.985**
6400	Raw	PSNR	31.98	-	-	-	-	-	-	-	43.08	43.30	**43.85**
		SSIM	0.765	-	-	-	-	-	-	-	0.985	0.986	**0.988**
	sRGB	PSNR	31.85	31.87	33.75	35.42	26.40	35.93	35.50	38.98	38.82	39.26	**39.99**
		SSIM	0.784	0.880	0.925	0.940	0.908	0.958	0.957	0.974	0.975	0.977	**0.980**
12800	Raw	PSNR	28.07	-	-	-	-	-	-	-	40.58	40.57	**41.20**
		SSIM	0.623	-	-	-	-	-	-	-	0.978	0.979	**0.982**
	sRGB	PSNR	29.27	29.79	31.59	33.54	25.54	34.91	35.23	37.47	37.54	37.80	**38.44**
		SSIM	0.688	0.778	0.902	0.910	0.908	0.952	0.9561	0.967	0.970	0.971	**0.975**
25600	Raw	PSNR	26.29	-	-	-	-	-	-	-	40.56	40.58	**41.17**
		SSIM	0.497	-	-	-	-	-	-	-	0.977	0.978	**0.982**
	sRGB	PSNR	25.53	25.95	29.48	30.52	26.24	33.64	34.63	35.26	35.28	35.51	**36.21**
		SSIM	0.495	0.559	0.868	0.835	0.904	0.942	0.952	0.957	0.960	0.960	**0.968**
Average	Raw	PSNR	32.01	-	-	-	-	-	-	-	43.25	43.37	**43.97**
		SSIM	0.732	-	-	-	-	-	-	-	0.984	0.985	**0.987**
	sRGB	PSNR	31.79	31.48	34.16	34.81	26.26	35.87	35.60	38.97	38.83	39.19	**39.95**
		SSIM	0.752	0.826	0.922	0.921	0.912	0.957	0.960	0.972	0.974	0.975	**0.979**

the noisy inputs.

3.2. User Study

Since there is no ground truth for outdoor videos, we conduct a user study to evaluate the denoising performance for our method, and two competing methods, i.e. EDVR [6] and DIDN [8]. For each scene, we randomly select a noisy video from the five ISO settings for the user study. Therefore, there are ten videos for ten dynamic scenes involved in the user study. For each scene, the three denoising results are randomly displayed and the workers are asked to vote for their preferred denoising results. There are 22 workers participating in the user study. The voting results are presented in Fig. 5. It can be observed that our method outperforms EDVR and DIDN for most videos. The results of DIDN are the worst since it does not consider the temporal correlations in denoising, which leads to large flicking artifacts. EDVR outperforms our result on the 8th and 10th videos since the two videos have large camera shaking and the workers are distracted by the camera motion and ignore the flicking artifacts.

References

[1] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. *CVPR*, 2019. 2

[2] Chen Chen, Qifeng Chen, Minh N. Do, and Vladlen Koltun. Seeing motion in the dark. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 1, 3

[3] Michele Claus and Jan van Gemert. Videnn: Deep blind video denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3

[4] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-

Figure 4. Comparison of the denoising results on the SMD dataset. The second and fourth rows present the close up for the regions highlighted by green boxes. For each row, from left to right: the noisy input with digital gain 10, SMD results with the digital gain used in their original paper, SMD results with digital gain 10, and our results with digital gain 10. Our results are directly generated *without retraining* on the SMD dataset.

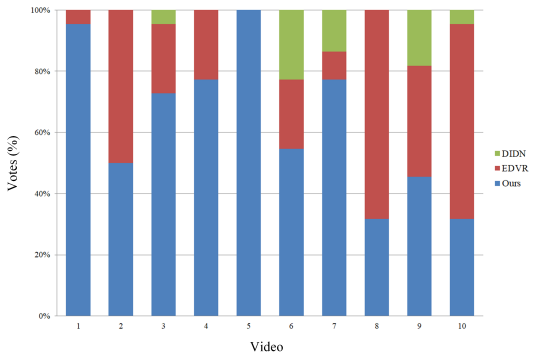

Figure 5. The user study results for ten outdoor videos. The video order is the same as that listed in Fig. 2.

domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 1

[5] Maggioni Matteo, Giacomo Boracchi, Foi Alessandro, Egiazarian Karen, et al. Video denoising using separable 4d nonlocal spatiotemporal transforms. In *Image Processing: Algorithms and Systems IX*, pages 1–11. SPIE, 2011. 3

[6] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3

[7] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 3

[8] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep iterative down-up cnn for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3