

# [Supplementary material] Self-supervision versus synthetic datasets: which is the lesser evil in the context of video denoising?

Valéry Dewil   Arnaud Barral   Gabriele Facciolo   Pablo Arias

Université Paris-Saclay, CNRS, ENS Paris-Saclay, Centre Borelli, 91190, Gif-sur-Yvette, France

<https://centreborelli.github.io/VDU2020-the-lesser-evil>

## 1. Unprocessing of sRGB dataset

Our *synthetic dataset* is synthesized from the REDS dataset [4]. It consists in clean sRGB videos with real motion. In order to create the *synthetic dataset*, we need two steps: unprocess back sRGB data to the raw domain and adding realistic noise. In this section, we talk about the unprocessing steps. We follow *Brooks et al.* [2], with some modifications to adapt it to our case. First the REDS dataset is made with 8-bits quantized frames. To reduce the effect of the quantization we add a quantization noise to each pixel value sampled from uniform distribution in the range  $[-1/2, 1/2]$ . Originally, the authors of [2] provide the *Camera Color Matrix* for four different cameras from the *Darmstadt Noise Dataset (DND)* [5]. In our case we want to simulate a single camera, thus we use only one of them.

The white balance is image dependent and thus inverting it is not straightforward. In [2], the authors estimated a range of realistic red and blue gains from *DND* (normalized with respect to the green gain being set to 1). They found that the red gain  $g^R$  has to be sampled uniformly in  $[1.9, 2.4]$  and the blue gain  $g^B$  in the range  $[1.5, 1.9]$ . They also consider a global gain  $g^{\text{global}}$  applied to all channels (to invert the brightness adjustment in the forward pipeline). This global gain is sampled from a Gaussian distribution  $\mathcal{N}(0.8, 0.1)$ . The total per-channel gain for channel  $c$  is then  $g^{\text{global}}/g^c$ . Occasionally the global gain can become greater than 1, which causes saturation later in the pipeline. This is wanted by *Brooks et al.* to create highlights and saturation. However, none of our *surrogate datasets* contains saturated areas, thus we prevent our per-channel gain to exceed one by sampling a global gain from a truncated Gaussian instead, clipping its value to one.

For each experiment, the clean *synthetic* raw dataset is tailored to model the *surrogate dataset*. We use the same Bayer pattern and we match the ranges of both datasets. For that purpose, we apply to the synthetic videos an affine tone mapping that maps the 1% and the 99% percentiles of the *synthetic* dataset to those of the *surrogate dataset*.

The next subsection describes how we generate the noisy

counterpart of the clean raw data.

## 2. Simulating realistic noise

Let  $\{u_i\}_I$  be the set of unprocessed clean data and  $\{\tilde{v}_j\}_J$  be a dataset of real noisy data (the *surrogate dataset*). Given the clean data  $\{u_i\}_I$  we can generate realistic noisy data  $\{v_i\}_I$  by applying the heteroscedastic Gaussian noise model. For that purpose, the steps to follow are:

(1) Estimate from  $\{\tilde{v}_j\}_J$  the parameters  $a$  and  $b$  of an heteroscedastic Gaussian noise model.

(2) Simulate a set of sequences with synthetic noise  $\{v_i\}_I$  where each  $v_i = u_i + n_i$  with  $n_i \sim \mathcal{N}(0, \sqrt{au_i + b})$ . The pairs of sequences  $(\{u_i\}_I, \{v_i\}_I)$  can then be used for training with supervision.

For addressing the point 1, we used two different strategies. For the Experiment I, we model a camera with a synthetic noise generator [7] and thus we can simulate the acquisition of flat-field images. Contrarily for the Experiment II and III, we want to model the noise model of a given camera having only a few noisy sequences. We followed two different methods to evaluate the noise model parameters. Both are described in the next subsection.

### 2.1. Noise parameters estimation

**Estimation for Experiment I.** In Experiment I we use the noise model introduced in [7], which models extreme low-light noise as a sum of a Poisson and a Tukey lambda distributions. In that sense, we have a simulated camera and the goal is to model its noise by a heteroscedastic Gaussian model whose variance is  $\sigma^2(u) = au + b$ , where  $u$  is the clean frame. To calibrate the  $a, b$  parameters, we simulate the acquisition of *flat-field* images, which is the usual way to calibrate signal dependent noise models.

We sample a range of constant patches  $P_i$  with intensity level  $i$ . For each patch  $P_i$  we generate a noisy patch  $\hat{P}_i$  using the Poisson-Tukey lambda noise model and compute the variance  $\sigma_i$  of the noisy patches. The parameters  $a$  and  $b$  are deduced from the points cloud  $(i, \sigma_i)$  using the least square error method. Figure 2 shows a plot of this points cloud and

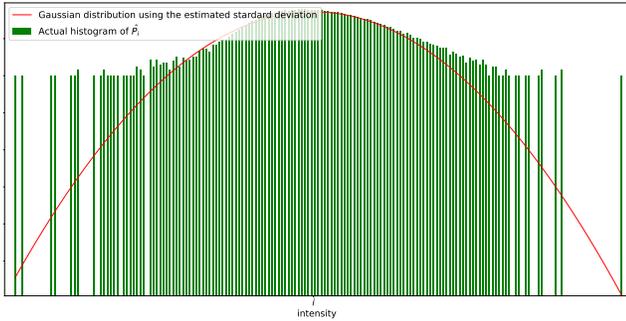


Figure 1. Example of actual histogram of the physic-based noise model ([7]) and the heteroscedastic Gaussian fitting (the  $y$ -axis is in logarithmic scale).

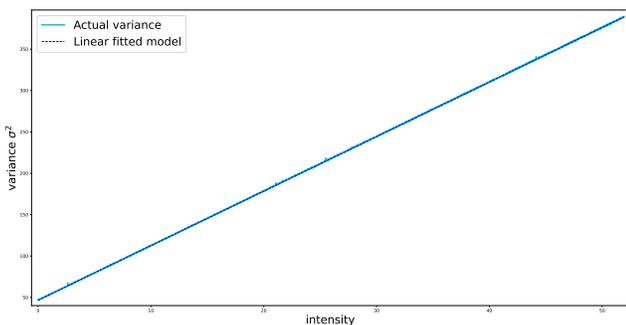


Figure 2. Variance of the noise model [7] and the estimated linear model.

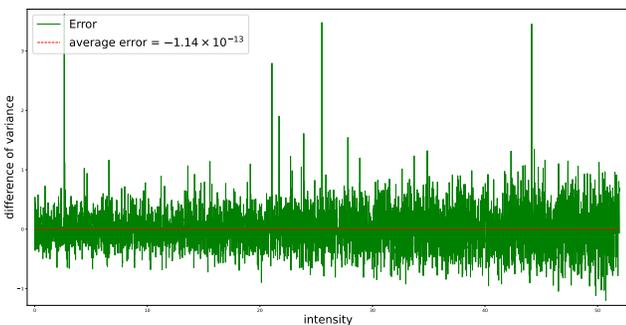


Figure 3. Difference of the variance of the noise model [7] and the estimated linear model.

the estimated linear model. The variance estimated from the Poisson-Tukey lambda noise is (as expected) an affine function of the intensity, therefore the affine model fits perfectly. Figure 3 shows the difference between the actual variance and the estimated linear model. We can see that the error is very small relatively to the variance value. The heteroscedastic Gaussian model will have the same intensity-variance curve, but the distributions are very different. Figure 1 shows the histogram of the variance for a patch of middle range intensity. The estimated Gaussian distribution

is also displayed. It can be seen that around the mean, the Poisson-Tukey lambda noise is well approximated by the Gaussian distribution. However, the Tukey lambda component has heavier tails than the Gaussian distribution.

**Estimation in the Experiments II and III.** In the case of Experiments II and III, the *surrogate* datasets consist of real noisy sequences but cannot generate more samples. Thus we need to estimate the camera noise level function (NLF) directly from the real data (SIDD [1] in Experiment II or CRVD [8] for the Experiment III). For that purpose, we estimate the NLF of each frame from each sequence of the *surrogate dataset* using the method of Ponomarenko [3, 6]. For each individual noisy frame  $v_i$ , this method estimates a set of intensity-variance points which are samples from the NLF. We gather estimated intensity-variance points of each frame into a large point cloud. Figure 4 shows this point cloud for Experiment II (one camera of the SIDD dataset). We use transparent points, thus the level of opacity gives an indication of the density in the point cloud. We then fit an affine model  $\sigma^2(u) = au + b$  by minimizing the least square error.

## 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*, pages 1692–1700, 2018. 2
- [2] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11036–11045, June 2019. 1
- [3] Miguel Colom and Antoni Buades. Analysis and Extension of the Ponomarenko et al. Method, Estimating a Noise Curve from a Single Image. *Image Processing On Line*, 3:173–197, 2013. 2
- [4] 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/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1
- [5] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1586–1595, 2017. 1
- [6] Nikolay N Ponomarenko, Vladimir V Lukin, MS Zriakhov, Arto Kaarna, and Jaakko Astola. An automatic approach to lossy compression of aviris images. In *2007 IEEE International Geoscience and Remote Sensing Symposium*, pages 472–475. IEEE, 2007. 2
- [7] Kaixuan Wei, Ying Fu, Yinqiang Zheng, and Jiaolong Yang. Physics-based noise modeling for extreme low-light photog-

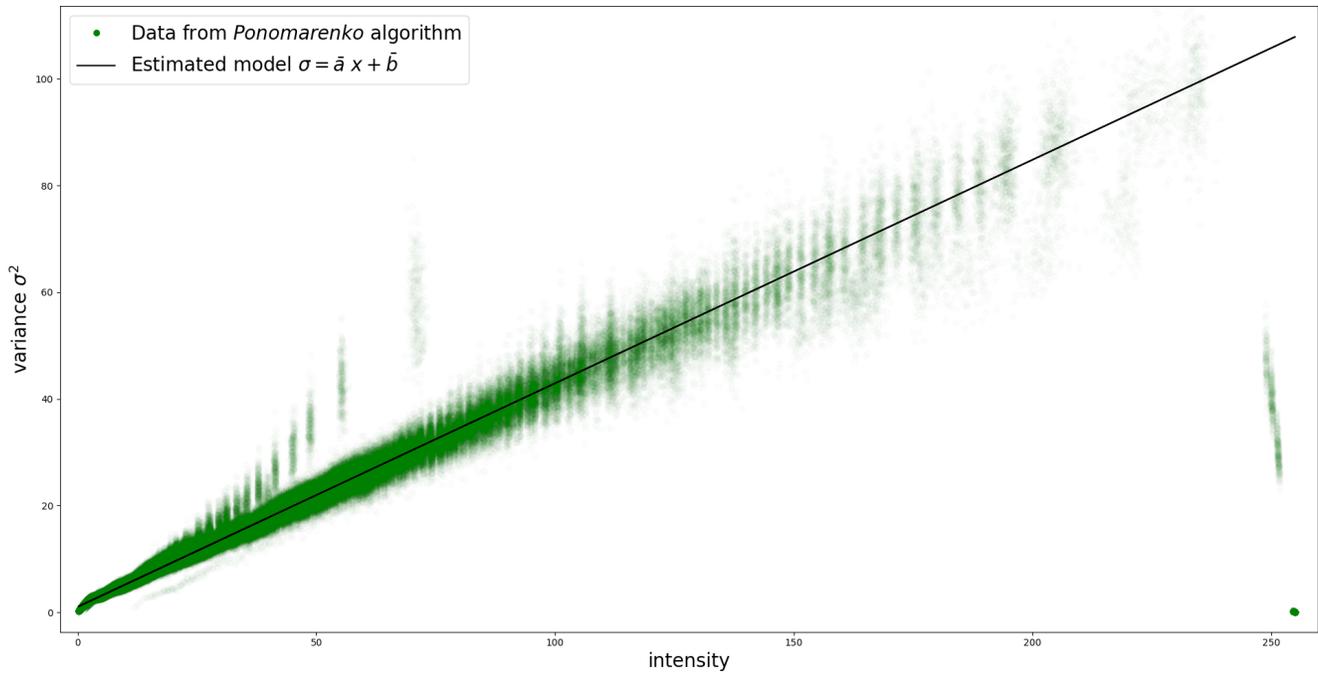


Figure 4. Linear model estimation of the noise curve.

raphy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1, 2

- [8] Huanjing Yue, Cong Cao, Lei Liao, Ronghe Chu, and Jingyu Yang. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2301–2310, June 2020. 2