FastDVDnet: Towards Real-Time Deep Video Denoising Without Flow Estimation
Supplementary Materials

Matias Tassano
GoPro France
mtassano@gopro.com

Julie Delon
MAP5, Université de Paris & IUF
julie.delon@parisdescartes.fr

Thomas Veit
GoPro France
tveit@gopro.com

1. Two-step denoising

FastDVDnet features a two-step cascaded architecture. The motivation behind this is to effectively employ the information existent in the temporal neighbors, and to enforce the temporal correlation of the remaining noise in output frames. To prove that the two-step denoising is a necessary feature, we conducted the following experiment: we modified a *Denoising Block* of FastDVDnet (see the associated paper) to take five frames as inputs instead of three, which we will refer to as *Den_Block_5inputs*. In this way, the same amount of temporal neighboring frames are considered and the same information as in FastDVDnet is processed by this new denoiser. A diagram of the architecture of this model is shown in Fig. 1. We then trained this new model and compared the results of denoising of sequences against the results of FastDVDnet.

Table 1 displays the PSNRs on four 854×480 color sequences for both denoisers. It can be observed that the cascaded architecture of FastDVDnet presents a clear advantage on *Den_Block_5inputs*, with an average difference of PSNRs of $0.95dB$. Additionally, results by *Den_Block_5inputs* present a sharp increase on temporal artifacts—flickering. Despite it being a multi-scale architecture, *Den_Block_5inputs* cannot handle the motion of objects in the sequences as well as the two-step architecture of FastDVDnet can. Overall, the two-step architecture shows superior performance with respect to the one-step architecture.

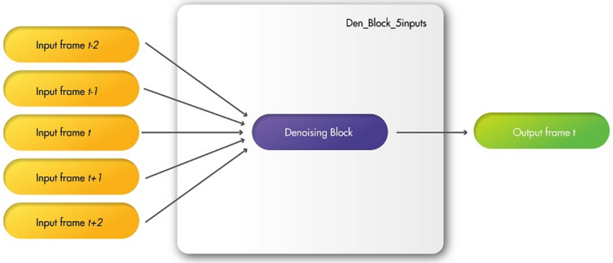

Figure 1. Architecture of the *Den_Block_5inputs* denoiser.

Table 1. Comparison of $PSNR$ of two denoisers on four sequences. Best results are shown in bold. Note: for this test in particular, neither of these denoisers implement residual learning.

		FastDVDnet	Den_Block_5inputs
$\sigma = 10$	hypersmooth	**37.34**	35.64
	motorbike	**34.86**	34.00
	rafting	**36.20**	34.61
	snowboard	**36.50**	34.27
$\sigma = 30$	hypersmooth	**32.17**	31.21
	motorbike	**29.16**	28.77
	rafting	**30.73**	30.03
	snowboard	**30.59**	29.67
$\sigma = 50$	hypersmooth	**29.77**	28.92
	motorbike	**26.51**	26.19
	rafting	**28.45**	27.88
	snowboard	**28.08**	27.37

2. Multi-scale architecture and end-to-end training

In order to investigate the importance of using multi-scale denoising blocks in our architecture, we conducted the following experiment: we modified the FastDVDnet architecture by replacing its *Denoising Blocks* by the denoising blocks of DVDnet. This results in a two-step cascaded architecture, with single-scale denoising blocks, trained end-to-end, and with no compensation of motion in the scene. We will call this new architecture FastDVDnet_Single. Table 2 shows the PSNRs on four 854×480 color sequences for both FastDVDnet and FastDVDnet_Single. It can be seen that the usage of multi-scale denoising blocks improves denoising results considerably. In particular, there is an average difference of PSNRs of $0.55dB$ in favor of the multi-scale architecture.

Table 2. Comparison of $PSNR$ of a single-scale denoiser against a multi-scale denoiser on four sequences. Best results are shown in bold. Note: for this test in particular, neither of these denoisers implement residual learning.

		FastDVDnet	FastDVDnet_Single
$\sigma = 10$	hypersmooth	**37.34**	36.61
	motorbike	**34.86**	34.30
	rafting	**36.20**	35.54
	snowboard	**36.50**	35.50
$\sigma = 30$	hypersmooth	**32.17**	31.54
	motorbike	**29.16**	28.82
	rafting	**30.73**	30.36
	snowboard	**30.59**	30.04
$\sigma = 50$	hypersmooth	**29.77**	29.14
	motorbike	**26.51**	26.22
	rafting	**28.45**	28.11
	snowboard	**28.08**	27.56

3. Ablation studies

A number of modifications with respect to the baseline architecture discussed in the associated paper have been tested, namely:

- the use of *Leaky ReLU* [8] or *ELU* [4] instead of *ReLU*. In neither case significant changes in performance were observed, with average differences in PSNR of less than $0.05dB$ on all the sequences and standard deviation of noise considered.

- optimizing with respect to the Huber loss [6] instead of the L_2 norm. No significant change of performance was observed. The mean difference in PSNR on all the sequences and standard deviation of noise considered was $0.04dB$ in favor of the L_2 norm case.

- removing batch normalization layers. An drop in performance of $0.18dB$ on average was observed for this case.

- taking more input frames. The baseline model was modified to take 7 and 9 input frames instead of 5. No improvement in performance was observed in neither case. It was also observed an increased difficulty of these models, which have more parameters, to converge during training with respect to the case with 5 input frames.

4. Upscaling layers

In the multi-scale denoising blocks, the upsampling in the decoder is performed with a *PixelShuffle* layer [9]. This layer repacks its input of dimension $4n_{ch} \times h/2 \times w/2$ into an output of size $n_{ch} \times h \times w$, where $ch,\ h,\ w$ are the number of channels, the height, and the width, respectively. In other words, this layer constructs all the 2×2 non-overlapping patches of its output with the pixels of different channels of the input, as shown in Fig. 2

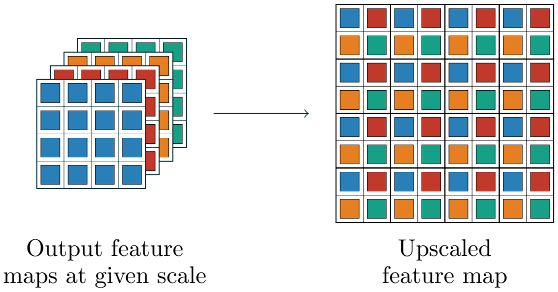

Figure 2. Upscaling layer.

5. Gaussian noise model

Recently, a number of algorithms have been proposed for video and burst denoising in low-light conditions, e.g. [3, 10, 7]. What is more, some of these works argue that real noise cannot be accurately modeled with a simple Gaussian model. Yet, the algorithm we propose here has been developed for Gaussian denoising because although Gaussian i.i.d. noise is not utterly realistic, it eases the comparison with other methods on comparable datasets—one of our primary goals. We believe Gaussian denoising is a middle ground where different denoising architectures can be compared fairly. Some networks which are proposed to denoise a specific low-light dataset are designed and overfitted given the image processing pipe of said dataset. In some cases, the comparison against other methods which have not been designed for the given dataset—e.g. the current version of our method—might not be accurate. Nonetheless, low-light denoising is not the main objective of our submission. Rather, it is to show that a simple, yet carefully designed architecture can outperform other more complex methods. We believe that the main challenge to denoising algorithms is the input signal-to-noise ratio. In this regard, the presented results have similar characteristics to low-light videos.

6. Permutation invariance

The algorithm proposed for burst deblurring and denoising in [1] features invariance to the permutation of the ordering of its input frames. One might be tempted to replicate its characteristics in an architecture such as ours to benefit from the advantages of the permutation invariance. However, the application of our algorithm is video denoising—which is not identical to burst denoising. Actually, the order in the input frames is a prior exploited by our algorithm to enforce the temporal coherence in the output sequence. In other words, permutation invariance is not necessarily desirable in our case.

7. Recursive processing

As previously discussed, in practice, the processing of our algorithm is limited to five input frames. Given this limitation, one would wonder if the theoretic performance bound might be lower to that of other solutions based on recursive processing (i.e. using the output frame in time t as input to the next frame in time $t+1$). Yet, our experience with recursive filtering of videos is that it is difficult for the latter methods to be on par with methods which employ multiple frames as input. Although, in theory, recursive methods are asymptotically more powerful in terms of denoising than multi-frame methods, in practice the performance of recursive methods suffers due to temporal artifacts. Any misalignment or motion compensation artifact which might appear in the output frame at a given time is very likely to appear in all subsequent outputs. An interesting example to illustrate this point is the comparison of the method in [5] versus the video non-local Bayes denoiser (VNLB [2]). The former implements a recursive version of VNLB, which results in a lower complexity algorithm, but with very inferior performance with respect to the latter.

References

[1] Miika Aittala and Frédo Durand. Burst image deblurring using permutation invariant convolutional neural networks. In *European Conference on Computer Vision*, pages 748–764. Springer International Publishing, 2018. 2

[2] Pablo Arias and Jean-Michel Morel. Video denoising via empirical bayesian estimation of space-time patches. *Journal of Mathematical Imaging and Vision*, 60(1):70–93, Jan 2018. 3

[3] C. Chen, Q. Chen, M. Do, and V. Koltun. Seeing motion in the dark. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3184–3193, 2019. 2

[4] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. 2

[5] T. Ehret, J. Morel, and P. Arias. Non-local kalman: A recursive video denoising algorithm. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3204–3208, 2018. 3

[6] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 2

[7] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Trans. Graph.*, 35(6), Nov. 2016. 2

[8] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013. 2

[9] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883. IEEE, Jun 2016. 2

[10] Wei Wang, Xin Chen, Cheng Yang, Xiang Li, Xuemei Hu, and Tao Yue. Enhancing low light videos by exploring high sensitivity camera noise. pages 4110–4118, 10 2019. 2