Patch Craft: Video Denoising by Deep Modeling and Patch Matching (Supplementary material)

Gregory Vaksman
CS Department - The Technion
Technion City, Haifa, Israel
grishav@campus.technion.ac.il

Michael Elad
Google Research
Mountain-View, California
melad@google.com

Peyman Milanfar
Google Research
Mountain-View, California
milanfar@google.com

1. Difference Between Patch-Craft Frames

As described in Section 2 of the paper, our algorithm augments each processed frame with nf patch-craft frames. We emphasize that *all these are different*, not identical nor shifted versions of each other. Thus, each brings an important additional information for the denoising to leverage. Here is an illustrative example to clarify this point. Assume for simplicity that $n = 1$, i.e., only one nearest neighbor is used. Consider two patch-craft frames with two different offsets, $[0, 0]$ and $[h_{offs}, v_{offs}]$, as shown in Figure 1.

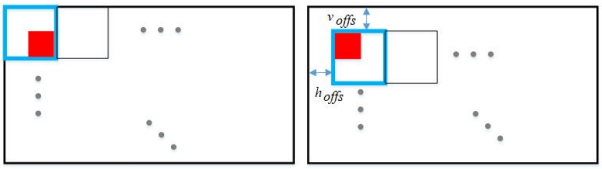


Figure 1: Inconsistency of patch overlaps in patch-craft frames with different offsets.

Consider the blue patches in these frames and their overlap red area. Each blue patch is a nearest neighbor (NN) of a corresponding patch in the processed frame, which means that the blue patches come from different locations in the video (perhaps even different frames). As such, their red regions are different, holding each additional information about the corresponding area of the processed frame. More broadly, all patch-craft frames are similar to each other but not identical, thus enriching the denoising process.

2. Time and Space Complexity

When evaluating complexity of a video denoiser, we should consider several different measures: (i) *Model-Size*: Number of learnable parameters; (ii) *Time-Complexity:* Number of operations per pixel; (iii) *Runtime:* Inference-wise; and (iv) *Training*: The complexity of training the network. We refer to each of these in details hereafter.

Model-Size: PaCNet has $2.87 \cdot 10^6$ trainable parameters.

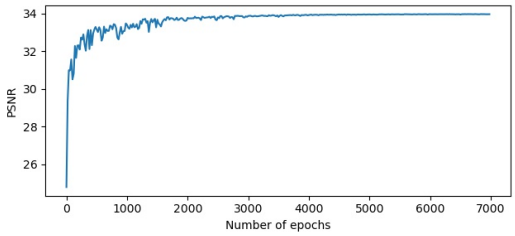
Time-Complexity: PaCNet consists of two stages: a NN search and a forward pass of the neural network. The forward pass performs $2.87 \cdot 10^6$ multiplications per pixel. The NN search in a bounding box of size $89 \times 89 \times 7$ with a patch size of 15×15 requires $(89-14)^2 \times 7 = 0.04 \times 10^6$ multiplications per pixel. Note that NN can be implemented such that it is independent of the patch size. Thus, in summary, PaCNet performs 2.91×10^6 multiplications per pixel.

Runtime: Our video denoising requires 30 seconds per frame. However, this timing is misleading, as the runtime is heavily dependent on hardware and software efficiency and implementation. In particular, our NN search is implemented highly inefficiently. In addition, the current implementation of separable convolutions on GPU is known to be unnecessarily slow. Both these problems can be overcome by a more careful implementation or a dedicated hardware for video processing.

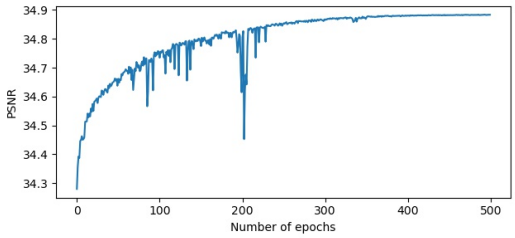
Training: We train the network for 7000 epochs, while each contains 90 randomly cropped training sample videos. Therefore, the total number of samples (with repetitions) used for training our model is $7000 \times 90 = 6.3 \times 10^5$.

3. Additional Details Regarding Training

Figures 2a and 2b present graphs of PSNR versus number of epochs during training of our networks. The values shown in the graphs are a rough estimation of training PSNR obtained by evaluating the networks on a small set of short videos randomly cropped from the training set. The spatial network, S-CNN, is trained using spatio-temporal $3D$ boxes of size $150 \times 150 \times 7$, applying denoising on the central frame of size $64 \times 64 \times 1$, where the rest of the box is used for nearest neighbor search. The boxes are randomly cropped from the training video seqiences. We use batches of size 10 and train the network for 7000 epochs. For training the temporal network, T-CNN, we use batches of 10 randomly cropped spatial-temporal boxes of size $64 \times 64 \times 7$ and run training for 500 epochs.



(a) S-CNN



(b) T-CNN

Figure 2: PSNR vs. the number of epochs for the validation set during training of the spatial and the temporal denoising networks, S-CNN and T-CNN, for noise level $\sigma = 30$. (We use different validation sets for each network).

4. Additional Results

Figure 4 presents graphs showing PSNR versus frame number for several test video sequences comparing PaCNet performance with VNLB [1], VNLnet [2], and FasDVDnet [3]. Figures 5, 6 and 7 show visual comparisons of our method versus leading algorithms. In addition to these figures, we attach to our paper several video (AVI) files that show comparisons of video sequences. Each file simultaneously plays the outcomes of four denoising algorithms: VNLB [1], VNLnet [2], FastDVDnet [3], and PaCNet (ours), along with the clean and the noisy sequences. These sequences are arranged according to the chart shown in Figure 3.

Files *salsa_s40_merge_rect.avi* and *skate-jump_s20_merge_rect.avi* show the video sequences *salsa* and *skate-jump* contaminated by noise with $\sigma = 40$ and $\sigma = 20$ respectively. There are two rectangles, red and green, in each video. The rest four files show zoom-in on the area in these rectangles:

- The green rectangle in *salsa* is shown in *salsa_s40_merge_zoom_g.avi*. As can be seen, our result is sharper than the VNLB outcome and less noisy than the outputs of FastDVDnet and VNLnet – see for example the floor. Also, observe that the VNLnet has noticeable artifacts around the legs.
- The red rectangle in *salsa* is shown in *salsa_s40_merge_zoom_r.avi*. As can be seen, PaCNet leads to better reconstruction – see for example the brick wall. Our output is sharper and less noisy than the competitors' results.
- The green and the red rectangles of *skate-jump* are shown in *skate-jump_s20_merge_zoom_g.avi* and *skate-jump_s20_merge_zoom_r.avi* respectively. As can be seen here as well, our algorithm leads to better reconstruction – e.g. see the trees.

Clean	Noisy	VNLB
VNLnet	FastDVDnet	PaCNet

Figure 3: Video chart.

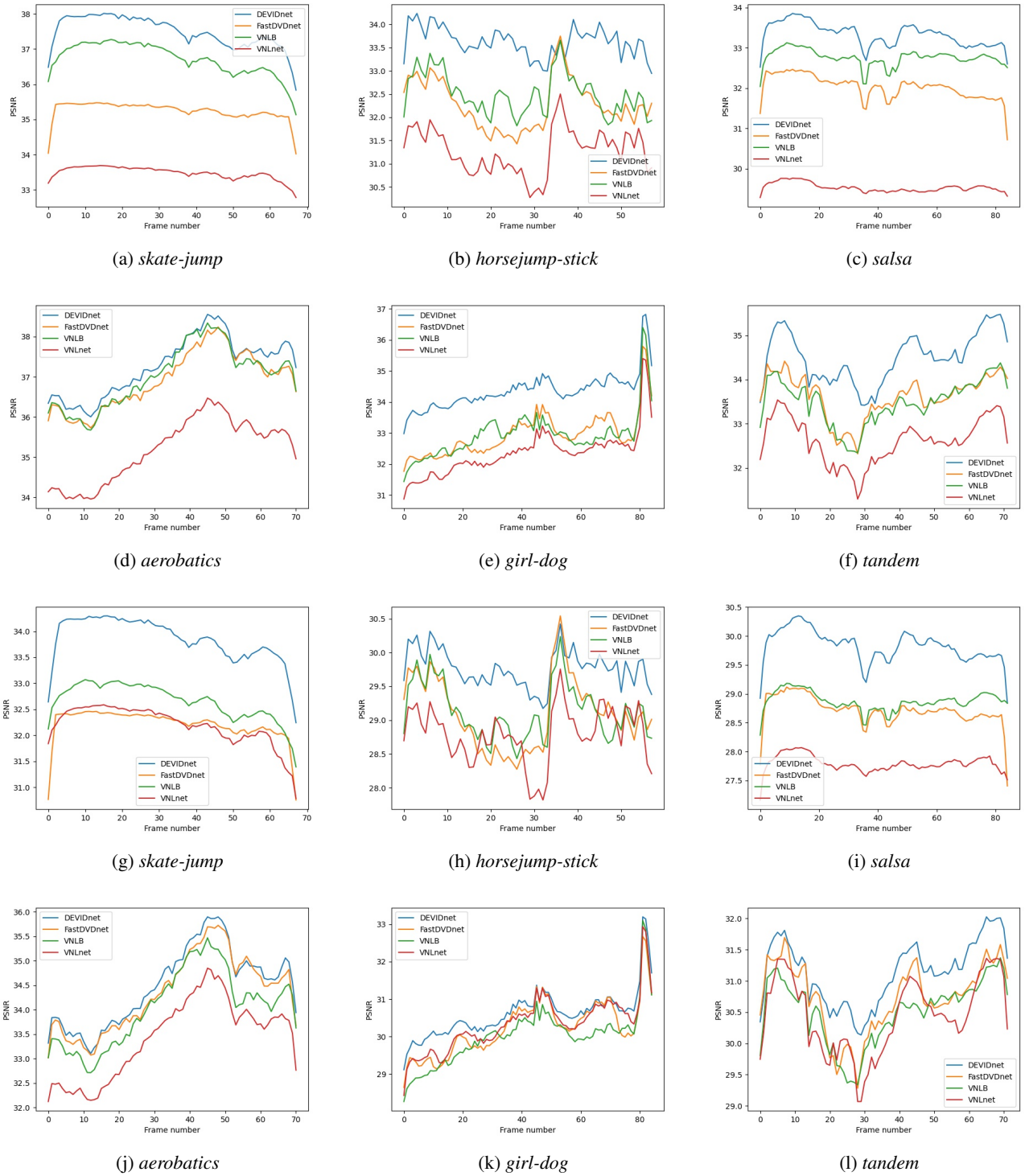
The videos are better seen in repeat mode.

References

[1] 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, 2018. 2, 4, 5, 6

[2] Axel Davy, Thibaud Ehret, Jean-Michel Morel, Pablo Arias, and Gabriele Facciolo. A non-local cnn for video denoising. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2409–2413. IEEE, 2019. 2, 4, 5, 6

[3] Matias Tassano, Julie Delon, and Thomas Veit. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1354–1363, 2020. 2, 4, 5, 6



(a) *skate-jump* (b) *horsejump-stick* (c) *salsa*

(d) *aerobatics* (e) *girl-dog* (f) *tandem*

(g) *skate-jump* (h) *horsejump-stick* (i) *salsa*

(j) *aerobatics* (k) *girl-dog* (l) *tandem*

Figure 4: PSNR vs. frame number for video sequences *skate-jump*, *horsejump-stick*, *salsa*, *aerobatics*, *girl-dog*, and *tandem*. The two first rows show denoising experiments with noise level $\sigma = 20$ and the third and fourth rows with $\sigma = 40$.

(a) Original (b) Noisy with $\sigma = 40$ (c) VNLB [1], PSNR = 29.14dB

(d) VNLnet [2], PSNR = 28.03dB (e) FastDVDnet [3], PSNR = 29.03dB (f) PaCNet (ours), PSNR = 30.15dB

(g) Original (h) Noisy with $\sigma = 40$ (i) VNLB [1], PSNR = 27.68dB

(j) VNLnet [2], PSNR = 26.69dB (k) FastDVDnet [3], PSNR = 27.56dB (l) PaCNet (ours), PSNR = 28.31dB

Figure 5: Denoising example with $\sigma = 40$. The figure shows frame 9 of the sequence *salsa*. The PSNR values appearing in 5c, 5d, 5e and 5f refer to the whole frame, whereas those in 5i, 5j, 5k and 5l refer to the cropped area. As can be seen, PaCNet leads to better reconstructed results – see the face and the details in the background building.

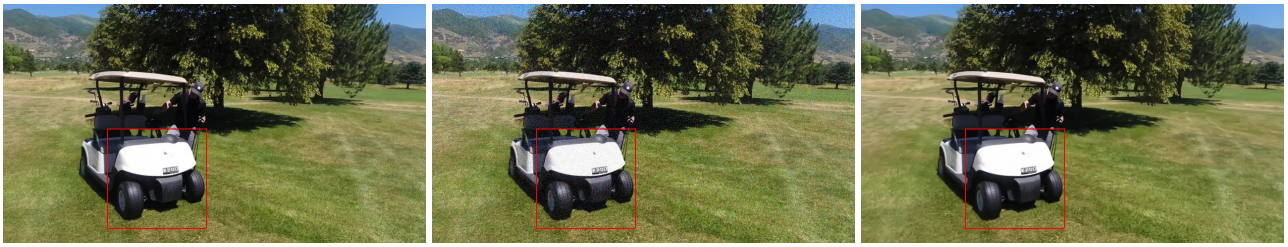
(a) Original (b) Noisy with $\sigma = 20$ (c) VNLB [1], PSNR = 34.18dB

(d) VNLnet [2], PSNR = 33.26dB (e) FastDVDnet [3], PSNR = 34.42dB (f) PaCNet (ours), PSNR = 34.74dB

(g) Original (h) Noisy with $\sigma = 20$ (i) VNLB [1], PSNR = 38.43dB

(j) VNLnet [2], PSNR = 37.71dB (k) FastDVDnet [3], PSNR = 38.53dB (l) PaCNet (ours), PSNR = 39.09dB

Figure 6: Denoising example with $\sigma = 20$. The figure shows frame 23 of the sequence *tractor*. The PSNR values appearing in 6c, 6d, 6e and 6f refer to the whole frame, whereas those in 6i, 6j, 6k and 6l refer to the cropped area. As can be seen, PaCNet leads to better reconstructed results – see the text on the trailer.



(a) Original (b) Noisy with $\sigma = 20$ (c) VNLB [1], PSNR = 31.22dB

(d) VNLnet [2], PSNR = 30.19dB (e) FastDVDnet [3], PSNR = 31.29dB (f) PaCNet (ours), PSNR = 32.14dB

(g) Original (h) Noisy with $\sigma = 20$ (i) VNLB [1], PSNR = 32.75dB

(j) VNLnet [2], PSNR = 31.15B (k) FastDVDnet [3], PSNR = 32.40dB (l) PaCNet (ours), PSNR = 33.42dB

Figure 7: Denoising example with $\sigma = 20$. The figure shows frame 18 of the sequence *golf*. The PSNR values appearing in 7c, 7d, 7e and 7f refer to the whole frame, whereas those in 7i, 7j, 7k and 7l refer to the cropped area. As can be seen, PaCNet leads to better reconstructed results – see the pattern on wheels.