LEARNING MODEL-BLIND TEMPORAL DENOISERS WITHOUT GROUND TRUTHS

Yanghao Li1 Bichuan Guo1 Jiangtao Wen1 Zhen Xia2 Shan Liu2 Yuxing Han3

1 Tsinghua University, Beijing, China 
2 Tencent Media Lab 
3 Research Institute of Tsinghua University in Shenzhen, Shenzhen, China

ABSTRACT

Denoisers trained with synthetic noises often fail to cope with the diversity of real noises, giving way to methods that can adapt to unknown noise without noise modeling or ground truth. Previous image-based method leads to noise overfitting if directly applied to temporal denoising, and has inadequate temporal information management especially in terms of occlusion and lighting variation. In this paper, we propose a general framework for temporal denoising that successfully addresses these challenges. A novel twin sampler assembles training data by decoupling inputs from targets without altering semantics, which not only solves the noise overfitting problem, but also generates better occlusion masks by checking optical flow consistency. Lighting variation is quantified based on the local similarity of aligned frames. Our method consistently outperforms the prior art by 0.6-3.2dB PSNR on multiple noises, datasets and network architectures. State-of-the-art results on reducing model-blind video noises are achieved.

Index Terms— temporal denoising, model-blind, optical flow

1. INTRODUCTION

Noise reduction is a crucial first step in video processing pipelines. Despite the steady advancements in sensor technology, visible noises still occur when recording in low lighting conditions [2] or on mobile devices [3]. Therefore, effective denoisers are essential for achieving satisfactory results in downstream applications [4, 5].

While there is a vast literature on reducing synthetic noises, reducing noises without explicit models (i.e. model-blind) remains an essential and challenging problem. On one hand, most traditional [6, 7, 8, 9] and data-driven methods [10, 11, 12] assume additive white Gaussian noise (AWGN). However, denoisers trained with synthetic AWGN often perform poorly on real noises [13]. On the other hand, creating training data by synthesizing all possible noises is computationally prohibitive and prone to bias [14, 15]. As a result, self-adaptive methods that do not require explicit noise modeling or expensive ground truths (GT) datasets have drawn considerable attention lately [16, 17, 18, 19].

This work was supported by the Tencent R&D Program (No. 61521002).

Fig. 1. An overview of existing and our methods. Notations: \( y_i \) (noisy frames), \( w^f/w^b \) (forward/backward flow), \( \hat{x}_i \) (denoised \( y_i \)), \( f_i \rightarrow j \) (\( f_i \) warped towards \( f_j \)). (a) Ehret et al. [1]. Training inputs and targets are constructed by aligning adjacent frames. (b) The direct extension of [1] to multi-frame inputs. Noise overfitting occurs due to pixels in \( y_1 \) appear both in inputs and targets. (c) Our method. By construction, any input and its target have no overlapping sources hence overfitting is avoided.

One of such methods is the recent FRAME2FRAME [1]. It is built upon the NOISE2NOISE framework [19], which trains an image denoising network with noisy-noisy pairs (as opposed to normally used noisy-clean pairs). By aligning video frames using optical flow, FRAME2FRAME constructs such pairs from the to-be-denoised video (Fig. 1(a)). These pairs are then used to train the denoiser.

Despite the pioneering work of FRAME2FRAME, its performance is limited due to the flowing problems: (i) It cannot be directly extended to temporal denoising: adjacent frames are included in both inputs and targets (Fig. 1(b)). This dual-presence causes overfitting in static regions (Fig. 2(left)). (ii) Occlusion and lighting variation are not properly handled when aligning the frames. As a result, the key assumption of NOISE2NOISE, that noisy-noisy pairs have the same GT, is easily violated.
In this paper we propose a novel twin sampler based framework for model-blind temporal denoising that successfully addresses all these problems, as is outlined in Fig. 1(c). The proposed framework:

- Decouples inputs from targets to include indirect temporal information from adjacent frames while preventing dual-presence.
- Utilizes optical flow consistency to provide occlusion mask and lighting variation map for better alignment management.

All these components are efficiently united under the proposed framework to deliver a boosted denoising performance.

2. METHODS

2.1. Background: The Key Assumption in NOISE2NOISE

The NOISE2NOISE framework assumes that, the noisy-noisy pairs \((y_i, y'_i)\) share the same GT:

\[ y_i = x_i + n_i, \quad y'_i = x_i + n'_i, \]

where \(n_i\) and \(n'_i\) are two independent noises on the same clean image \(x_i\). If a denoiser network \(g_\theta\) with weights \(\theta\) is trained by minimizing the empirical risk:

\[ \arg\min_\theta \mathbb{E}_{y_i, y'_i}[\ell(g_\theta(y_i), y'_i)], \]

where \(\ell\) is, say, \(L_2\) loss, it learns to approximate the optimal estimator \(g^*\), which is \(\mathbb{E}[y'_i \mid y_i]\) according to Bayesian decision theory. If the noise distribution further satisfies

\[ \mathbb{E}[y'_i \mid y_i] = \mathbb{E}[x_i \mid y_i], \]

i.e. the noise \(n'_i\) preserves mean, the optimal estimator \(g^*\) would appear as if the network was trained using noisy-clean pairs \((y_i, x_i)\). The same property holds for \(L_1/L_0\) loss under median/mode-preserving noises.

2.2. Twin Sampler

The twin sampler serves as the core of the proposed framework. It not only includes the indirect temporal information from adjacent frames while avoiding the dual-presence, but also uses backward optical flow to construct free training data.

Intuitively, suppose the denoiser \(g_\theta\) originally takes three adjacent frames \(Y_2 = \{y_1, y_2, y_3\}\) as input to denoise the middle frame \(y_2\). We warp \(y_2\) to align with \(y_3\), yielding \(y_{2\rightarrow3}\), and replace it with \(y_3\). The new input is \(Y'_2 = \{y_1, y_2, y_{2\rightarrow3}\}\), and the target is still \(y_{3\rightarrow2}\). The key is that the new input \(Y'_2\) and the target \(y_{3\rightarrow2}\) do not share sources: pixels in \(Y'_2\) originate from \(y_1\) and \(y_2\), and pixels in \(y_{3\rightarrow2}\) originate from \(y_3\). As such, a degenerated mapping that produces part of the input will not be learned. Also, since \(Y'_2\) keeps the semantic form of the original input \(Y_2\), no change is required during inference time. As a free byproduct, another noisy pair, \((Y'_1 = \{y_1, y_3, y_4\}, y_{2\rightarrow3})\), can be immediately constructed without additional computation.

Formally, Suppose the denoiser network input is

\[ Y_i := \{\ldots, y_{i-1}, y_i, y_{i+1}, \ldots\}, \]

and an optical flow estimator \(\Gamma\) computes the optical flow from \(a\) to \(b\) as \(\Gamma(a, b)\). The forward and backward flow \(w^f\) and \(w^b\) are computed between the estimated \(\hat{x}_{i-1}\) and \(\hat{x}_i\) as

\[ \hat{x}_{i-1} := g_\theta(Y_{i-1}), \quad \hat{x}_i := g_\theta(Y_i), \]

\[ \hat{w}^f := \Gamma(\hat{x}_{i-1}, \hat{x}_i), \quad \hat{w}^b := \Gamma(\hat{x}_i, \hat{x}_{i-1}). \]

Then the two noisy pairs are constructed as

\[ \begin{align*}
Y'_{i-1} &= Y_{i-1} \setminus \{y_i\} \cup \{y_{(i-1)\rightarrow i}\}, \quad y_i \rightarrow (i-1), \\
Y'_i &= Y_i \setminus \{y_{i-1}\} \cup \{y_{(i-\rightarrow i-1)}\}, \quad y_{i-\rightarrow i-1}.
\end{align*} \]

where \(y_i \rightarrow j\) is the frame obtained by warping \(y_i\) towards \(y_j\).

2.3. Alignment Management

In the case of occlusion or lighting changes, the NOISE2NOISE assumption in (3) may fail. In FRAME2FRAME, occlusion is detected by checking if the divergence of optical flow exceeds a threshold. We derive a better occlusion mask by testing the forward-backward consistency assumption stated in [20], using the forward and backward optical flow computed in the previous subsection (see Fig. 3 (d) and (e)). Specifically, let \(p\) denote a pixel coordinate in \(y_i\). We compute a binary map \(o_i\) to mark if \(p\) is occluded in the previous frame \(y_{i-1}\): \(o_i(p) := 0\) (not occluded) if

\[ \|w^b(p) + w^f(p + w^b(p))\|_2^2 < \alpha_1 \left(\|w^b(p)\|_2 + \|w^f(p + w^b(p))\|_2\right) + \alpha_2, \]

otherwise \(o_i(p) := 1\) (occluded), where \(\alpha_1, \alpha_2\) are hyperparameters specifying relative and absolute thresholds.

Lighting variation is usually quantified by the difference between corresponding pixels, e.g. pixels at same coordinates of \(x_i\) and \(x'_i (x_{i-1} \text{ warped towards } x_i)\). However, individual pixels can have large variance. To improve robustness, we instead compare the average intensity of patches centered at
Fig. 3. An illustration of alignment management: (a) frame $y_{i-1}$; (b) frame $y_i$; (f) $y'_i$ (frame $y_{i-1}$ warped towards $y_i$); (d) inferred occlusion mask (solid black) and lighting variation (gray) from Sec. 2.3; (c) and (g): multiply (b) and (f) with mask (d), respectively; (e) inferred occlusion mask based on flow divergence; (h) GT occlusion. Compare (b) and (f) to observe occlusion and lighting variation. Further compare with (c) and (g) to see that the mask (d) effectively covers these outlier pixels.

corresponding pixels. Using a $5 \times 5$ box filter $\kappa_5$, the patch difference can be computed as $|\kappa_5 \ast (x_i - x'_i)|$, where $\ast$ denotes convolution. Again, occluded pixels should be excluded from the patch. Formally, the lighting variation $l_i$ of pixels in $x_i$ with respect to corresponding pixels in $x_{i-1}$ is

$$l_i := \frac{|\kappa_5 \ast (\hat{x}_i - \hat{x}'_i) \circ (1 - o_i)|}{\kappa_5 \ast (1 - o_i) + \epsilon},$$

where $\hat{x}'_i := \text{warp}(\hat{x}_{i-1}, w^b)$ is a warped version of $\hat{x}_i$ in (5), $\circ$ denotes point-wise product, and the denominator is a normalization factor that contains a small positive $\epsilon = 10^{-6}$ to prevent division by zero.

2.4. Training Losses

The occlusion map $o_i$ and lighting variation $l_i$ are used to adjust the loss $l$ in (2). For noisy pair (8), its loss is

$$\ell(g_0(Y'_i) \circ \gamma, y_{(i-1)\rightarrow i} \circ \gamma)$$

where $\gamma = (1 - o_i) \circ \xi(l_i)$, and $\xi(l) := \exp(-\alpha_3 l)$ is used to map its input to $(0, 1)$ with hyper-parameter $\alpha_3$. This loss function ensures that occluded pixels do not contribute to the loss, and pixels with drastic lighting variation contribute less to the loss. Loss for noisy pair (7) comes similarly.

The pseudocode summarizing the above procedures is shown in Algorithm 1.

3. EXPERIMENTAL RESULTS

3.1. Data and Implementation Details

Data preprocessing. We use synthetic noises for quantitative experiments as in [1], and demonstrate real noise reduction visually (Fig.2(right)). Five distinct synthetic noises are used for testing: AWGN20 (AWGN with standard deviation $\sigma=20$), MG (multiplicative Gaussian, where each pixel’s value is multiplied by a $\mathcal{N}(1, 0.3^2)$ Gaussian), CG (correlated Gaussian, where AWGN with $\sigma=25$ is blurred with a $3 \times 3$ box filter), IR (impulse random, where each pixel has 10% chance to be replaced by a uniform random variable in $[0, 255]$), and JPEG (JPEG compressed Gaussian, where each frame is compressed with 60% JPEG quality after adding AWGN with $\sigma=25$). To mimic realistic scenarios, all pixel values are clipped to range $[0, 255]$ and rounded to integers.

We collect clean videos from three datasets: Sintel [25], DAVIS [26] and Derf’s collection [27]. The “clean” pass of Sintel training set (23 sequences) are split into 11:4:8, which are used for optical flow training (sintel-tr), hyper-parameter tuning (sintel-val) and denoising performance evaluation (sintel-eval), respectively. All 30 sequences from the “test-dev” split of DAVIS (davis-tr) and 7 selected sequences [28] from Derf’s collection (derf-7) are also used for performance evaluation.

Implementation. To demonstrate the generality of our framework, we apply it to latest video denoising networks VNLNet [28] and FastDVDnet [29]. The weight used to initialize VNLNet is the publicly released version trained on color sequences with AWGN. The authors of FastDVDnet included noise strength in network input for non-blind denoising. We train a blind version by removing the noise strength input and repeating the same training procedure.

We perform random search to determine the best hyper-parameters. A “validation noise” (AWGN with $\sigma=30$) is used to prevent previous “test noises” from being seen. The combination that achieves the best average PSNR on sintel-val is: $\alpha_1=0.0064$, $\alpha_2=1.4$ in (9) and $\alpha_3=5.0$ in (11). The loss function $\ell$ in (2) is the L1 loss, which can cope with a wide
Table 1. Average PSNR/SSIM on derf-7 and davis-30. DnCNN+f2f is the original implementation of [1]. FastDVDnet+f2f and VNLnet+f2f are direct extensions of [1] to multi-frame input. “X initial” is the initial model of X pretrained with AWGN, “X+ours” is our proposed framework applied to X.

| dataset | noise | AVGN20 | MG | IR | JPEG | AVGN20 | MG | IR | JPEG |
|---------|-------|--------|----|----|------|--------|----|----|------|
| derf-7  |       |        |    |    |      |        |    |    |      |
| VBM4D [21] | 35.25/896 | 20.11/850 | 22.70/471 | 27.55/743 | 29.44/793 | 32.79/890 | 27.28/801 | 22.44/439 | 26.92/719 | 28.77/767 |
| VNLB [22] | 34.71/916 | 22.16/589 | 23.20/509 | 21.56/498 | 30.55/852 | 34.00/911 | 19.44/474 | 23.32/509 | 21.61/514 | 30.28/853 |
| VDNet [23] | 33.05/893 | 20.22/463 | 22.04/452 | 19.64/387 | 23.73/486 | 33.58/912 | 18.13/380 | 22.11/446 | 19.90/399 | 23.92/479 |
| CBDNet [15] | 31.91/866 | 24.12/646 | 24.19/582 | 24.76/613 | 27.84/717 | 32.45/890 | 21.95/564 | 24.62/598 | 26.39/682 | 28.56/754 |
| ViDeNN [14] | 33.51/903 | 20.08/460 | 22.53/476 | 18.12/329 | 23.90/492 | 34.37/924 | 19.78/379 | 22.54/471 | 18.15/320 | 23.99/477 |
| TOFlow [24] | 32.89/884 | 23.92/652 | 23.49/646 | 27.65/786 | 24.85/740 | 31.02/854 | 23.16/558 | 23.49/632 | 25.69/703 | 24.73/730 |
| DnCNN+f2f | 31.97/874 | 28.82/815 | 27.24/735 | 29.68/830 | 29.90/826 | 31.38/870 | 26.72/757 | 27.26/747 | 28.70/795 | 29.64/828 |
| VNLnet+ours | 34.89/928 | 22.00/579 | 23.93/573 | 20.89/446 | 28.28/733 | 34.67/927 | 19.28/465 | 24.07/571 | 21.07/467 | 28.32/721 |
| VNLnet+f2f | 28.41/743 | 26.73/715 | 25.40/624 | 29.90/818 | 27.52/709 | 28.87/792 | 25.99/714 | 25.81/669 | 29.02/801 | 27.96/765 |
| VNLnet+ours | 34.89/928 | 30.24/849 | 30.40/844 | 31.34/838 | 31.13/867 | 34.67/927 | 27.81/780 | 29.57/822 | 30.84/860 | 30.48/854 |
| FastDVDnet+ours | 35.26/904 | 22.44/361 | 23.03/304 | 21.83/485 | 27.95/705 | 34.39/927 | 19.81/445 | 23.23/508 | 22.03/507 | 28.44/710 |
| FastDVDnet+ours | 30.55/839 | 28.23/779 | 26.08/673 | 29.54/817 | 29.06/792 | 30.07/847 | 26.78/772 | 26.63/726 | 28.72/805 | 28.81/810 |

Table 2. Average PSNR/SSIM on sintel-8. “ts”: twin sampler. If twin sampler is disabled, the direct extension of FRAME2FRAME is used. “occ”: occlusion inference method. “div”/“ofc” : occlusion is inferred based on optical flow divergence/consistency. “l” : lighting variation. If it is disabled, lighting variations $I_i$ are set to 0. “od”: online denoising.

| components | sintel-8 |        |        |        |
|------------|---------|--------|--------|--------|
| x | y | z | ts | oc | cr | div | ofc | l | od |
| AVGN20 | 30.70 | 27.56 | 25.30 | 30.01 | 29.12 | 30.45 | 31.11 |
| AVGN40 | 30.24 | 27.32 | 25.03 | 29.75 | 28.92 | 30.45 | 31.11 |
| AVGN60 | 30.00 | 27.11 | 24.50 | 29.55 | 28.72 | 30.45 | 31.11 |
| AVGN80 | 29.78 | 26.92 | 24.00 | 29.39 | 28.50 | 30.45 | 31.11 |

3.2. Main Results

Regarding overall performance, we primarily compare with FRAME2FRAME, which uses the image denoiser DnCNN as their backbone (initialized by pretraining with AWGN σ=20). The direct extension of FRAME2FRAME to multi-frame input also serves as a baseline. Traditional methods such as VBM4D and VNLB, as well as some recent blind denoising methods including CBDNet and ViDeNN are also compared. Since our task is model-blind denoising, using specialized pretrained model for each test noise is not allowed. Therefore, for methods that require pretrained weights, the same publicly released model will be used for all noises.

Table 1 shows the overall results on derf-7 and davis-30. More details are given in the table caption. From the results, we clearly see that: (1) Due to the overfitting problem, direct extensions of FRAME2FRAME perform even worse than its DnCNN-based version (compare rows with suffix “+f2f”), as shown in Fig. 2. (2) Our method consistently outperforms DnCNN-based FRAME2FRAME on both architectures, achieving 0.6-3.2dB PSNR gain (“DnCNN+f2f” v.s. rows with suffix “+ours”). (3) Comparing to other existing methods, our method achieves state-of-the-art results on removing model-blind noises.

3.3. Ablation Studies

Table 2 shows the detailed breakdown of our method’s performance on dataset sintel-8. The twin sampler offers the most significant contribution, as PSNR is improved by 1.1-3.7dB (row 1 v.s. 2). For occlusion masking, flow consistency clearly outperforms flow divergence, achieving 0.4-0.7dB PSNR gain (row 2 v.s. 3). By considering lighting variation, PSNR is improved by 0.06-0.2dB (row 3 v.s. 4). Online denoising provides 0.1-0.4dB PSNR gain in total (row 4 v.s. 5). To demonstrate our method’s robustness to noise levels, Table 2 also lists two different Gaussian noises: AWGN20 (σ=20) and AWGN40 (σ=40). It can be seen that our proposed method achieves consistent improvement across different noise strengths.

4. CONCLUSION

We present a general framework model-blind denoising without clean signals. The twin sampler not only resolves the overfitting problem suffered by the direct extension of image-based methods, but also operates efficiently by reusing estimated optical flow. The rest components further boost denoising performance via occlusion masking and lighting variation penalty. Our results indicate that a video denoiser should look at frame differences and similarities simultaneously: noise attributes can be learned from the former, while temporal information can be extracted from the latter. Our method consistently outperforms the prior art by 0.6-3.2dB PSNR on multiple noises and datasets.
5. REFERENCES

[1] T. Ehret, A. Davy, J.M. Morel, G. Facciolo, and P. Arias, “Model-blind video denoising via frame-to-frame training,” in CVPR, 2019.

[2] C. Chen, Q. Chen, J. Xu, and V. Koltun, “Learning to see in the dark,” in CVPR, 2018.

[3] A. Abdelhamed, S. Lin, and Michael. S. Brown, “A high-quality denoising dataset for smartphone cameras,” in CVPR, 2018.

[4] D. Liu, B. Cheng, Z. Wang, H. Zhang, and T. S. Huang, “Enhance visual recognition under adverse conditions via deep networks,” IEEE TIP, 2019.

[5] D. Liu, B. Wen, X. Liu, Z. Wang, and T. S. Huang, “When image denoising meets high-level vision tasks: a deep learning approach,” in IJCAI, 2018.

[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” IEEE TIP, 2007.

[7] W. Dong, L. Zhang, G. Shi, and X. Li, “Nonlocally centralized sparse representation for image restoration,” IEEE TIP, 2012.

[8] S. Gu, L. Zhang, W. Zuo, and X. Feng, “Weighted nuclear norm minimization with application to image denoising,” in CVPR, 2014.

[9] D. Zoran and Y. Weiss, “From learning models of natural image patches to whole image restoration,” in ICCV, 2011.

[10] Y. Chen and T. Pock, “Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration,” IEEE TPAMI, 2016.

[11] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising,” IEEE TIP, 2017.

[12] K. Zhang, W. Zuo, and L. Zhang, “FFDNet: Toward a fast and flexible solution for CNN-based image denoising,” IEEE TIP, 2018.

[13] T. Plötz and S. Roth, “Benchmarking denoising algorithms with real photographs,” in CVPR, 2017.

[14] M. Claus and J. van Gemert, “ViDeNN: Deep blind video denoising,” in CVPR Workshops, 2019.

[15] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, “Toward convolutional blind denoising of real photographs,” in CVPR, 2019.

[16] J. Batson and L. Royer, “Noise2self: Blind denoising by self-supervision,” in ICML, 2019.

[17] T. Ehret, A. Davy, P. Arias, and G. Facciolo, “Joint denoising and denoising by overfitting of bursts of raw images,” ICCV, 2019.

[18] A. Krull, T. O. Buchholz, and F. Jug, “Noise2void – learning denoising from single noisy images,” in CVPR, 2019.

[19] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, “Noise2noise: Learning image restoration without clean data,” in ICML, 2018.

[20] N. Sundaram, T. Brox, and K. Keutzer, “Dense point trajectories by GPU-accelerated large displacement optical flow,” in ECCV, 2010.

[21] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, “Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms,” IEEE TIP, 2012.

[22] P. Arias and J. M. Morel, “Towards a Bayesian video denoising method,” in International Conference on Advanced Concepts for Intelligent Vision Systems, 2015.

[23] Z. Yue, H. Yong, Q. Zhao, L. Zhang, and D. Meng, “Variational denoising network: Toward blind noise modeling and removal,” in NeurIPS, 2019.

[24] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, “Video enhancement with task-oriented flow,” IJCV, 2019.

[25] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, “A naturalistic open source movie for optical flow evaluation,” in ECCV, 2012.

[26] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, “A benchmark dataset and evaluation methodology for video object segmentation,” in CVPR, 2016.

[27] Xiph.org, Derf’s Test Media Collection, 2019, https://media.xiph.org/video/derf/.

[28] A. Davy, T. Ehret, J. M. Morel, P. Arias, and G. Facciolo, “A non-local CNN for video denoising,” in ICIP, 2019.

[29] M. Tassano, J. Delon, and T. Veit, “FastDVDnet: Towards real-time video denoising without explicit motion estimation,” arXiv preprint arXiv:1907.01361, 2019.

[30] D. Sun, X. Yang, M. Liu, and J. Kautz, “PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume,” in CVPR, 2018.