Lightweight Video Denoising using Aggregated Shifted Window Attention

Lydia Lindner [1]
lydia.lindner@icg.tugraz.at

Alexander Effland [2]
effland@iam.uni-bonn.de

Filip Ilic [1]
filip.ilic@icg.tugraz.at

Thomas Pock [1]
thomas.pock@tugraz.at

Erich Kobler [2]
erich.kobler@ukbonn.de

[1] Graz University of Technology
[2] University of Bonn

Abstract

Video denoising is a fundamental problem in numerous computer vision applications. State-of-the-art attention-based denoising methods typically yield good results, but require vast amounts of GPU memory and usually suffer from very long computation times. Especially in the field of restoring digitized high-resolution historic films, these techniques are not applicable in practice. To overcome these issues, we introduce a lightweight video denoising network that combines efficient axial-coronal-sagittal (ACS) convolutions with a novel shifted window attention formulation (ASwin), which is based on the memory-efficient aggregation of self- and cross-attention across video frames. We numerically validate the performance and efficiency of our approach on synthetic Gaussian noise. Moreover, we train our network as a general-purpose blind denoising model for real-world videos, using a realistic noise synthesis pipeline to generate clean-noisy video pairs. A user study and non-reference quality assessment prove that our method outperforms the state-of-the-art on real-world historic videos in terms of denoising performance and temporal consistency.

1. Introduction

Image/video denoising and restoration have been the subject of research for several decades. This persistent focus of the research community is driven by the fact that denoising is essential for numerous image/video processing and computer vision tasks, e.g., for the reconstruction of microscopy images, tomography, or satellite data. In this work, we focus on denoising of digitized high-resolution historic films, which are degraded by various noise sources such as digital noise or film grain, which strongly depends on the facilitated film stock, the acquisition process, and the digitizing procedure [8, 36]. Typically, historic movies are recorded on analog film reels, which is why the noise of movie frames is only spatially correlated, not temporally. The spatial correlation of the noise is proportional to the resolution of digitized films [8], which results in highly correlated noise in digitized cinematic movies. Throughout this work, we refer to this type of noise as analog noise. Moreover, the high resolution of cinematic movies limits the complexity of denoising algorithms, despite increasing computing power and memory. Consequently, memory-efficient and fast video denoising models are required that deal with spatially correlated noise.

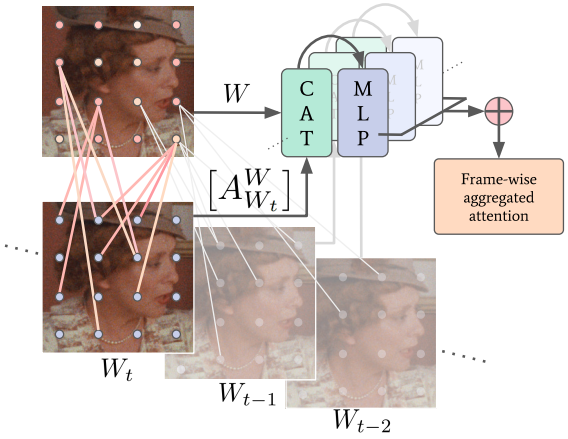


Figure 1: Lightweight frame-wise aggregated shifted window (ASwin) attention, designed for effective video denoising. At any position in the video, a multilayer perceptron (MLP) processes the concatenated (CAT) local information and an attention output, thereby efficiently fusing self- and cross-attention between frames.

Video denoising algorithms essentially exploit two principles – locality and self-similarity. The locality principle assumes that neighboring elements (pixels or frames) are more likely similar, while self-similarity [44] accounts for repetitive structures (e.g., edges, textures, objects) within an image/frame or between multiple frames. Image filtering methods [14, 32] and convolutional neural networks

https://github.com/LLindn/ASwin-Video-Denoising

(CNNs) [20,29,31] strongly exploit the locality principle by means of convolutions. Self-similarity-based methods build upon the extraction of similar repetitive structures, which are jointly processed, e.g., block-matching 4D collaborative filtering (BM4D) [25], video denoising via spatio-temporal nonlocal Bayes [2], or recently patch craft [33]. Interestingly, the extraction of patches and the collaborative filtering strongly resembles the query-key-value search pattern of transformers [34], which has outperformed the state-of-the-art in image and video processing [5,24]. Further, transformer and attention mechanisms have proven to be particularly suited for video processing [11,23]. In particular, the state-of-the-art in image denoising is obtained by combining convolutions and transformer blocks [42]. In this paper, we combine the locality and self-similarity principles within a novel deep learning architecture designed for effective video denoising. In detail, the main contributions of this paper are as follows:

- We propose a novel attention mechanism, termed *aggregated shifted window attention (ASwin)*, that exploits a frame-wise aggregated self- and cross-attention scheme with shared projection matrices, which is combined with a shifted window approach.
- For our attention mechanism, we utilize a frame-wise search window. In combination with axial-coronal-sagittal (ACS) convolutions [37] within our deep learning model, we obtain a lightweight (small memory usage and runtime) yet effective approach.
- We train our network as a general-purpose blind denoising model, suitable for real world noisy video data. A user study and a non-reference quality assessment shows that our method outperforms other state-of-the-art denoising approaches in terms of denoising performance and temporal consistency.

2. Related Work

Traditional video denoising algorithms pursue a spatial and temporal patching scheme to exploit redundancy of videos (self-similarity). For example, BM4D [25] extends the collaborative filtering idea of BM3D [7] to spatio-temporal patches and enforces sparsity in a higher-dimensional transform domain. Similarly, VNLB [2] performs a joint empirical Bayes estimation for each group, assuming a Gaussian model. Nowadays, these methods are typically outperformed by data-driven methods based on learned CNNs [21, 27, 30, 31, 43]. VNLnet [9] finds self-similar video patches using a nonlocal search algorithm, which are subsequently processed by a CNN. Recently, PaCNet [33] combined the concept of self-similarity and CNNs by means of artificial patch-craft frames, constructed by stacking matched patches. Furthermore, the query-key-value search principle of transformers [34] enables video models [11, 23] to incorporate self-similarity. VRT [23] applies this principle on multiple scales to extract long-range dependencies within videos and implements a warping scheme for motion compensation. For video denoising, VRT is conditioned on the noise level and thus its performance depends on an a-priori knowledge or estimation of the noise level. Other non-blind video denoising methods include DVDNet [30], FastDVDNet [31], and PaCNet [33]. In contrast, blind video denoising approaches [6, 26, 29, 39, 41, 42] do not require a noise level estimation and are therefore more suitable for real-world scenarios with unknown noise types and levels.

We can further classify methods based on their usage of motion information. For instance, DVDnet [30] incorporates optical flow in a three stage process. First, input frames are processed separately by an image denoising CNN, then the optical flow between frames is calculated using DeepFlow [35] to apply motion compensation, and finally the motion-compensated frames are processed by another CNN. FastDVDnet [31] extends DVDnet, however, it employs an end-to-end-trained UNet structure [28] that uses five consecutive input frames to reconstruct the central frame without explicitly accounting for the optical flow. The recent transformer VRT [23] performs explicit motion compensation by feature warping on different resolutions. However, motion compensation always bears the risk of introducing motion artifacts due to inaccurate optical flow estimation, which is especially noteworthy in the case of real-world noise that is often spatially correlated. Hence, we refrain from explicitly using optical flow within our model.

Regarding the style of learning, real-world approaches can be classified into unsupervised/self-supervised and supervised methods. The unsupervised/self-supervised framework exploits basic principles of natural images and videos to train a model with the goal of reconstructing corrupted parts of the data by using local neighborhood information [10, 22, 29]. Multi Frame-to-Frame (MF2F) [10] and Unsupervised Deep Video Denoising (UDVD) [29] are state-of-the-art in self-supervised video denoising and have been shown to perform well on removing real-world noise in videos. MF2F is fine-tuning a pre-trained FastDVDnet model by minimizing the distance to motion-compensated adjacent frames. This method incorporates TV-ℓ_1 optical flow [40], thereby creating a significant dependence on the accuracy of a motion estimate. This typically leads to blurry results [12, 38, 41]. UDVD avoids motion compensation by extending the blind spot framework [19] to video denoising using a causal rotated CNN. In contrast, supervised learning on synthesized realistic noisy-clean video pairs enables training of general-purpose blind video denoising models, as demonstrated for images [42].

3. Method

In this section, we introduce a novel attention mechanism termed *aggregated shifted window (ASwin) attention* that is combined with an efficient CNN, leading to a lightweight convolution-transformer model for video denoising. Formally, video denoising means restoring a clean video $\bar{u} \in \mathbb{R}^{F\times M\times N\times C}$ from a noisy observation $u \in \mathbb{R}^{F\times M\times N\times C}$, where F refers to the number of frames – each of size $M \times N$ with C channels. The relation between u and $\bar{u}$ is determined by the noise generation process.

3.1. Aggregated Shifted Window Attention

The proposed aggregated shifted window (ASwin) attention extends recent attention mechanisms [11, 24, 34] to effectively process video data. Attention layers are the core unit of a transformer, in which all elements in an input sequence of length L aggregate information from all other elements in parallel, thereby generating context information. Attention combines queries $Q \in \mathbb{R}^{L\times C}$, keys $K \in \mathbb{R}^{L\times C}$, and values $V \in \mathbb{R}^{L\times L}$ for $\mathbb{R}^C$-dimensional features of an input sequence. Attention is computed as the weighted sum of the values

$$\mathrm{SoftMax}(QK^\top)V,$$

where the matrix $QK^\top$ represents the similarity between the query-key pairs and the $\mathrm{SoftMax}$ function is applied along the rows. Note that the memory consumption increases quadratically in L. A straightforward adaption of recent 2D image processing transformers, by simply replacing convolutions and attention mechanisms with their respective 3D counterparts, results in models with large memory consumption and long computation times, which is therefore infeasible for high-resolution digitized analog films (e.g. cinematic scenes). There are several options to reduce memory requirements, e.g., restricting to a local window instead of a global computation [24] or chunking the queries [13]. In this work, we advance these ideas by introducing a frame-based aggregation scheme, where the attention is computed individually for all frames within local windows. Typically, non-overlapping windows are considered to reduce computation time and memory consumption. However, this could potentially lead to block artifacts, which can be circumvented by the shifted window approach (Swin) [24]. Let $[\![A]\!] := \{1, \dots, A\}$ for any $A \in \mathbb{N}$. We consider a fixed rectangular window $\overline{W} \subset \Omega := [\![F]\!] \times [\![M]\!] \times [\![N]\!]$. The key $\mathbf{k}_{t,h,w}$ and value $\mathbf{v}_{t,h,w}$ pair within a position $(t, h, w) \in \overline{W}$ are computed as

$$\mathbf{k}_{t,h,w} = P^K \mathbf{x}_{t,h,w} + \mathbf{b}^K \in \mathbb{R}^R,$$
$$\mathbf{v}_{t,h,w} = P^V \mathbf{x}_{t,h,w} + \mathbf{b}^V \in \mathbb{R}^C,$$

where $P^K \in \mathbb{R}^{R\times C}$, $P^V \in \mathbb{R}^{C\times C}$ are learned projection matrices, $\mathbf{b}^K \in \mathbb{R}^R$, $\mathbf{b}^V \in \mathbb{R}^C$ are learned biases, and $\mathbf{x}_{t,h,w} \in \mathbb{R}^C$ is the corresponding input feature vector in the window W. By stacking the resulting vectors, we obtain the subsequent matrices:

$$K_{\overline{W}} = \begin{pmatrix} \vdots \\ \mathbf{k}_{t,h,w}^\top \\ \vdots \end{pmatrix}, \; V_{\overline{W}} = \begin{pmatrix} \vdots \\ \mathbf{v}_{t,h,w}^\top \\ \vdots \end{pmatrix} \text{ for } (t, h, w) \in \overline{W}.$$

For any position $(f, m, n) \in W \subset \Omega$, which is not necessarily in $\overline{W}$, the query is defined as

$$\mathbf{q}_{f,m,n} = P^Q \mathbf{x}_{f,m,n} + \mathbf{b}^Q \in \mathbb{R}^R,$$

where $P^Q \in \mathbb{R}^{R\times C}$ and $\mathbf{b}^Q \in \mathbb{R}^R$ are learned as above. Likewise, the queries are stacked into the matrix

$$Q_W = \begin{pmatrix} \vdots \\ \mathbf{q}_{f,m,n}^\top \\ \vdots \end{pmatrix} \text{ for } (f, m, n) \in W.$$

Then, the weighted attention of two windows reads as

$$A_{\overline{W}}^W = \mathrm{SoftMax}(Q_W K_{\overline{W}}^\top / \sqrt{R}) V_{\overline{W}}, \tag{1}$$

where $\mathrm{SoftMax}$ denotes the row-wise softmax function. For clarity, the derivation only describes a single attention head. However, we do use multiple heads [34] in the implementation to simultaneously focus on different aspects within one transformer block.

We use the attention mechanism (1) to compute self-attention within a frame and cross-attention to adjacent video frames. Both windows are equal in the case of frame-wise self-attention ($W = \overline{W}$), whereas, for cross-attention between frames $\overline{W}$ is equal to W shifted along the frame dimension, i.e., only a temporal offset is applied. For the sake of simplicity, we denote this shifted window by W_t for $t \in [\![T]\!]$. Consequently, we get $\overline{W} = W_t$ in the case of cross-attention. To account for positions within the shifted windows, we utilize a 2D sine positional encoding.

To combine self- and cross-attention efficiently, we propose the following residual aggregation scheme. By starting from a feature $\mathbf{x}_{f,m,n} \in \mathbb{R}^C$ at position $(f, m, n) \in W \subset \Omega$ ASwin attention aggregates different temporal windows by

$$\mathbf{y}_{f,m,n} = \mathbf{x}_{f,m,n} + \frac{1}{|\mathcal{T}|} \sum_{t\in\mathcal{T}} h(\mathbf{x}_{f,m,n}, [A_{W_t}^W]_{f,m,n}),$$

where $\mathcal{T}$ is the set of considered temporal shifts, and $[A_{W_t}^W]_{f,m,n} \in \mathbb{R}^C$ denotes the corresponding row-vector of the attention $A_{W_t}^W$ at the considered location (f, m, n). Note that $f \in \mathcal{T}$ implies that self-attention is included. The fusion function $h\colon \mathbb{R}^C \times \mathbb{R}^C \to \mathbb{R}^C$ consists of a linear layer used for feature reduction of the channel-wise concatenation of $\mathbf{x}_{f,m,n}$ and $[A_{W_t}^W]_{f,m,n}$, followed by LayerNorm [3]

and a subsequent multilayer perceptron (MLP). The final attention $\mathbf{z}_{f,m,n}$ is obtained by function $g\colon \mathbb{R}^C \times \mathbb{R}^C \to \mathbb{R}^C$, consisting of the channel-wise concatenation of the initial feature $\mathbf{x}_{f,m,n}$ and frame-aggregated attention $\mathbf{y}_{f,m,n}$, followed by LayerNorm and another MLP.

$$\mathbf{z}_{f,m,n} = g(\mathbf{x}_{f,m,n}, \mathbf{y}_{f,m,n})$$

We highlight that the same fusion network h is used to aggregate the attention for all considered temporal frame shifts $t \in \mathcal{T}$. Consequently, ASwin requires only constant memory to process T frames.

3.2. Architecture

Our general network architecture is inspired by Swin-Conv-UNet (SCUNet) [42] and is depicted in Figure 2. Like SCUNet, our denoising strategy combines the local modeling ability of residual convolutions with the non-local modeling ability of efficient shifted window attention. Our architecture incorporates the proposed ASwin/ACSconv block as the main processing block of a residual UNet.

Instead of using standard 3D convolutions, which are expensive in terms of computation time and memory consumption, we incorporate ACS (axial-coronal-sagittal) convolutions [37], which aim to approximate standard 3D convolutions by splitting each kernel into three 2D parts and extract 2D spatial information along all three different axes pairs of a 3D volume. In particular, ACS convolutions can be seen as a special case of 3D convolution with blocks of sparse kernels. Since ACS kernels are reshaped 2D kernels, the number of learned parameters coincides with the number for 2D convolutions, which thus reduces memory consumption and computation time. The ACS block consists of a $3\times3\times3$ ACS convolution, followed by a ReLU activation and a second $3 \times 3 \times 3$ ACS convolution.

Each ASwin/ACSconv block consists of three residual stages in which the feature volume is first processed by a $1 \times 1 \times 1$ convolution, to achieve inter-channel communication, split evenly into two parts along the channel dimension after which one part is fed to the ASwin block and one to the ACSconv block, respectively. Finally, the residual of the input features is obtained by merging the processed output of each of the two blocks via concatenation along the channel dimension and post-processing by another $1\times1\times1$ convolution to allow an information flow between both blocks. The downsampling in the encoder part is obtained by a 2×2 convolution with spatial stride 2 and the upsampling in the decoder part is obtained in the same manner by a 2×2 transposed convolution with spatial stride 2. Temporal downsampling did not show to be beneficial during our initial experiment and is therefore omitted.

3.3. Real-world Noise Synthesis

Our proposed method is also designed for deep blind real-world video denoising, where we particularly focus on analog noise in digitized historic films. Since no ground truth exists for real digitized videos, the generation of synthetic realistic data of clean-noisy video pairs properly representing the distribution of real image noise (including analog noise) is necessary. For this reason, we modify the image noise synthesis pipeline presented in [42] for realistic video noise synthesis.

The main idea is based on the degradation of images by adding many various kinds of noise and including a resizing operation in order to approximate non-i.i.d. noise distributions commonly seen in digitized videos. The noise synthesis procedure builds upon a double degradation strategy with a random order of applying different noise models, which helps the generalization ability of the blind denoising model by further expanding the learned degradation space. The noise synthesis pipeline includes the following degradations, which are applied twice in random order: Gaussian noise, Poisson noise, camera sensor noise, speckle noise, jpeg compression noise, resizing. In particular, Gaussian noise is applied with a probability of 1 while all other degradations are applied with a probability of 0.5. The exact hyperparameters used for each degradation model can be found in the supplementary material.

We adopt the 3D generalized zero-mean Gaussian noise model with varying correlation across the color channels, with the two extreme cases being grayscale Gaussian noise and additive white Gaussian color noise. Signal-dependent (color or grayscale) Poisson noise is added to the clean video to simulate photon shot noise. Although we focus on digitized analog videos, modeling camera sensor noise is still of interest, since during the digitizing process the analog video can be processed in a similar manner as in a digital in-camera image processing pipeline (ISP). This kind of noise is incorporated by applying a reverse ISP pipeline [4] to the video, resulting in raw images. Subsequently, read and shot noise are added before applying the forward ISP pipeline to again obtain RGB images. Multiplicative speckle noise can simply be modeled by multiplying Gaussian noise (generated by Gaussian noise synthesis as above) to a clean image. Since JPEG compression causes reduced image quality and can introduce strong block artifacts, it is also considered in the noise synthesis. Digitized analog videos often exhibit analog film grain, which is spatially correlated noise. The resizing operation itself does not introduce any additional noise to clean videos, however, the noise distribution of a video already degraded with one of the noise models described above is altered. Upsampling leads to a higher spatial correlation of noise in the data, while downsampling can reduce the spatial correlation.

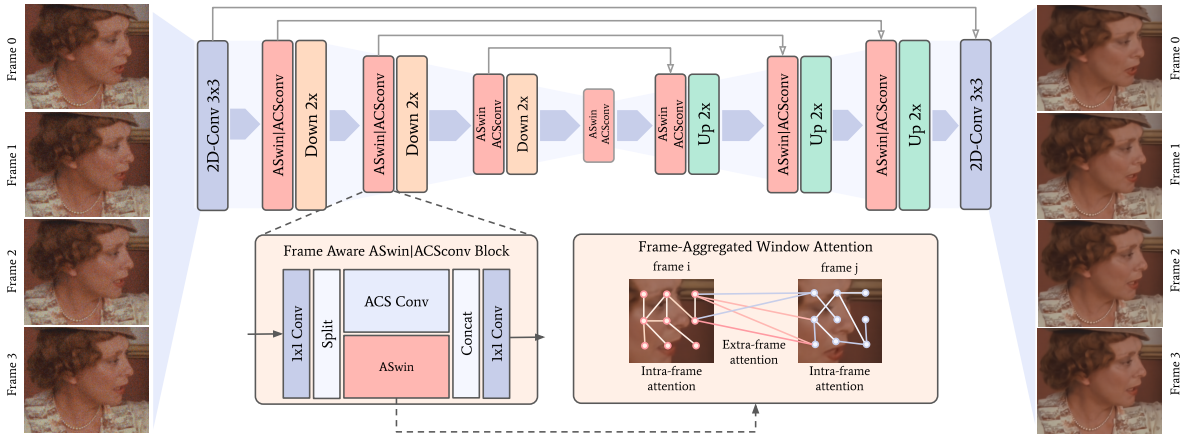


Figure 2: T noisy input frames are denoised in parallel. The model combines ASwin/ACSconv blocks as the main building blocks of a residual UNet. For details regarding the architecture see Section 3.2.

4. Experiments and Results

In this section, we present numerical details demonstrating the applicability and superiority of the proposed method. First, we elaborate on further details about the model configuration in Section 4.1. Subsequently, we benchmark our approach with competing methods on additive Gaussian noise in Section 4.2 and the challenging task of digitized analog film denoising in Section 4.3.

4.1. Training and Evaluation Setup

Our network consists of four scales, i.e., three encoder blocks, a body, and three decoder blocks with residual skip connections between the contracting and expanding path. We set the number of residual ASwin-ACSconv stages within each scale to 3, to effectively combine locality and (global) similarity. The 3D attention window size ($t \times h \times w$) of the ASwin blocks is set to $3 \times 8 \times 8$ in the first and second scale, and $6 \times 16 \times 16$ in the third and fourth scale, respectively. The number of channels is 64 in the first scale, 128 in the second scale, and 256 in the third and fourth scale.

We train our model on the DAVIS 2017 training data set [17]. In particular, we train on sequences of 6 frames, randomly cropped to 128×128 pixels. The network is trained for $50\,000$ epochs with a batch size of 4, using the Adam [18] optimizer to minimize the mean-squared error between our model prediction and the corresponding target. The initial learning rate is set to 10^{-4} and is decreased by a factor of 0.5 every $10\,000$ epochs.

For non-blind Gaussian denoising, we synthesize noisy/target pairs by simply adding white Gaussian noise to samples from the standard 480p DAVIS 2017 training set [17]. In contrast, for the task of blind real-world denoising, we synthesize corrupted noisy videos as described in Section 3.3. Here, we use the full DAVIS 2017 data set in high resolution, due to the downsampling operations in the noise synthesis. In detail, we randomly crop samples of size 600×600 pixels before noise synthesis. Then, the resulting noisy video frames are again randomly cropped to 128×128 pixels and fed to the denoising network for training. It is important to note that the synthetic noise data was generated on the fly during training, to further increase the variety of samples and expand the learned degradation space of the network. A random collection of generated noisy-clean videos can be found in the supplementary material. In the case of non-blind Gaussian denoising, we learn individual models for each noise level $\sigma \in \{10, 20, 30, 40, 50\}$, whereas for real-world denoising we train one general blind-denoising model applicable to any noisy video. For both scenarios, the number of denoised output frames equals the number of corrupted input frames.

During inference, we process 24 video frames in parallel to effectively exploit temporal redundancy. In detail, we divide each test video into groups of 24 frames such that neighboring groups overlap by 2 frames. Each group is processed individually, yielding the final predictions for the non-overlapping frames. For the overlapping frames, we calculate the mean to fuse both predictions.

4.2. Gaussian Denoising

Although not specifically designed for additive Gaussian denoising, we evaluate our approach on two commonly used data sets for synthetic denoising: Set8 [30] and DAVIS2017 [17]. A qualitative comparison of the results on a single frame of Set8 is shown in Figure 4. Zooming in on the snowboarder, visual differences become apparent, with PaCNet [33] displaying artifacts on the sky, FastDVDnet [31] preserving fewer fine details, such as the valleys of

the mountain in the background, and VRT [23] exhibiting a seemingly over-smoothing behavior on the yellow jacket.

In order to obtain a quantitative comparison to the state-of-the-art methods, we evaluate the denoising performance in terms of PSNR; the results are shown in Table 1. Our model yields results close to the state-of-the-art VRT [23] and consistently outperforms all other competing methods. When considering both the denoising performance and the runtime (see Figure 3), we can observe that the better performance of VRT comes along with a significant increase in computation time. In detail, VRT is 21.2× slower than our model. Moreover, VRT suffers from an excessive memory consumption and videos can only be processed in a patched way. We provide a detailed comparison of the memory consumption of VRT and our method in the supplementary material. The long runtime and memory restrictions, as well as the fact that VRT is designed as a non-blind denoising network, discard it as a candidate for real-world denoising on high resolution cinematic scenes. The same holds for PaCNet, which is the slowest of the compared learning-based methods. FastDVDnet has a low runtime, however, it performs significantly worse than our method, on both Davis and Set8, regardless of the noise level. Our model provides by far the best trade-off between performance and runtime, and is therefore very well applicable for the task of high-resolution cinematic video denoising.

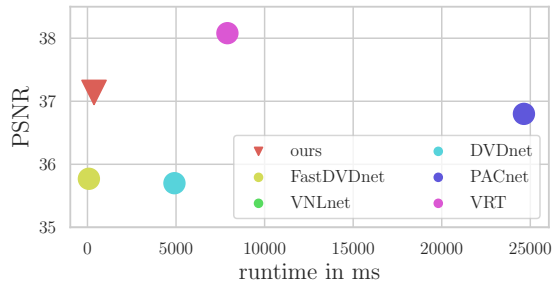


Figure 3: Visualization of denoising performance versus runtime for our method and other state-of-the-art methods.

4.3. Real-world Denoising

In this section, we show quantitative and qualitative results of our blind real-world denoiser and compare them to state-of-the-art real-world video denoising techniques. To evaluate the different methods, we used 10 high-resolution sequences of digitized analog film footage, exhibiting different unknown noise types of varying strength. The camera model, the analog film type, and the exact digitizing process are unknown. Further details and a visual overview of the real-world test data can be found in the supplementary. Denoising digitized analog videos is even more challenging

		VBM4D	VNLB	DVDnet	VNLnet	FastDVD	PaCNet	VRT	ours
	device	CPU	CPU	GPU	GPU	GPU	GPU	GPU	GPU
	runtime (s)	156.0	420.0	4.91	1.87	**0.08**	24.64	7.86	0.37
DAVIS	$\sigma = 10$	37.58	38.85	38.13	35.83	38.71	39.97	**40.82**	40.15
	$\sigma = 20$	33.88	35.68	35.70	34.49	35.77	36.82	**38.15**	37.12
	$\sigma = 30$	31.65	33.73	34.08	32.86	34.04	34.79	**36.52**	35.37
	$\sigma = 40$	30.05	32.32	32.86	32.32	32.82	33.34	**35.32**	34.13
	$\sigma = 50$	28.80	31.13	31.90	31.43	31.86	32.20	**34.36**	33.17
	mean	32.39	34.34	34.53	33.39	34.64	35.42	**37.03**	35.99
Set8	$\sigma = 10$	36.05	37.26	36.08	37.10	36.44	37.06	**37.88**	36.99
	$\sigma = 20$	32.19	33.72	33.49	33.88	33.43	33.94	**35.02**	34.06
	$\sigma = 30$	30.00	31.74	31.68	31.59	31.68	32.05	**33.35**	32.41
	$\sigma = 40$	28.48	30.39	30.46	30.55	30.46	30.70	**32.15**	31.22
	$\sigma = 50$	27.33	29.24	29.53	29.47	29.53	29.66	**31.22**	30.31
	mean	30.81	32.47	32.25	32.52	32.31	32.68	**33.92**	33.00

Table 1: Quantitative (PSNR) results for Gaussian denoising. The best and second best scores are printed in **bold** and blue. The runtime is given for a video frame of resolution 960 × 540 using FP16 precision.

than other real-world denoising tasks, due to a high spatial correlation of noise induced by the physical structure of analog film and additional digital noise caused by the digitizing process. We compare our approach to the state-of-the-art real-world denoising methods MF2F [10] and UDVD [29], which – due to operating in a self-supervised manner – were both fine-tuned directly on each noisy test video. We additionally compare our method to commercial denoising software for high-end video restoration, namely NeatVideo [1] and DarkEnergy [15].

4.3.1 Visual Quality Assessment

A qualitative comparison of our method, MF2F, UDVD, NeatVideo, and DarkEnergy is provided in Figure 5. A visual assessment shows that our method outperforms all other methods in terms of noise removal and detail preservation. In the first row of Figure 5, one can see that there is severe residual noise for all methods except ours, especially in bright areas. Our denoising algorithm is able to remove the noise completely while still preserving detail. The second and third row of Figure 5 show two other examples of the test data set. As can be seen, our approach again outperforms all other methods, which either create visually unpleasing artifacts, are too blurry, or are not able to remove the noise effectively. Since UDVD is based on a blind-spot denoising strategy (missing central pixel in the receptive field), the resulting denoised image is of low quality, due to a strong spatial correlation of the noise. MF2F generates visually more appealing results, however, it suffers from low temporal consistency, which is also confirmed by the results of the user study shown in section 4.3. An obvious drawback of fine-tuning a model during inference, as it is the case for UDVD and MF2F, is the largely increased runtime. In addition, MF2F requires an upfront optical flow estimation and generation of occlusion masks, which is a time-consuming task, especially in the case of high-resolution videos. UDVD does not depend on the prior estimation of motion, however, if the network is trained from scratch on a

Figure 4: Comparison of the qualitative denoising performance on Gaussian noise with $\sigma = 40$ on a test video of Set8.

single sequence, the time for the model to converge is considerable. Moreover, manual early stopping had to be performed to avoid overfitting to the noisy reference, due to spatially correlated noise in the videos. We observed that the commercial denoising software DarkEnergy often generates blob artifacts and produces slightly blurry results in general. In contrast, NeatVideo is not able to remove all the noise sufficiently, however, the details are well preserved and the overall result looks more visually appealing. Additional results can be found in the supplementary.

4.3.2 No-Reference Video Quality Assessment

Since the actual ground truth is not available for real-world noisy videos, standard quality assessment metrics, such as PSNR, cannot be computed. To get quantitative proof of the superiority of our approach compared to other state-of-the-art methods, we perform a No-Reference Image Quality Assessment (NR-IQA) on the denoised real videos. The goal of NR-IQA is to estimate the perceptual image or video quality in accordance with quality ratings provided by human subjects. Therefore, we use the state-of-the-art NR-IQA metric MUSIQ [16], which is computed by a multi-scale image quality transformer. Due to the multi-scale representation, this method can assess the visual quality at different granularities and outperforms traditional NR-IQA methods by a large margin. A high MUSIQ score refers to a high-quality image that is visually pleasing to a human. As can be seen in Table 2, the NR-IQA evaluation reinforces that our method is able to generate the highest quality reconstruction on the test videos. Detailed results of each video are provided in the supplementary.

	noisy	UDVD	MF2F	DarkEnergy	NeatVideo	ours
mean	25.11	25.77	35.29	31.05	33.14	**38.16**

Table 2: Quantitative evaluation of image quality using MUSIQ [16]. Mean over test data set. Best and second best score are printed in **bold** and blue, respectively.

4.3.3 User Study

To verify that the proposed method generates – besides numerically competitive results – also visually appealing videos, we conducted a user study including 30 individuals. In the user study design, we balanced the participants regarding their computer vision background. In detail, approximately 47% of the participants had no background in computer vision or related fields, while 53% of the participants had a computer vision background. The subjects were given an instruction sheet, explaining the tasks they were asked to perform (see supplementary material). The user study was conducted using 20 video sequences, which were obtained by taking two crops of each video from our real test data set and the respective denoised results. The user interface showed the reference video and three competing methods, which were anonymized by labeling them as "A", "B", and "C", as well as randomly shuffled w.r.t. the assigned labels to avoid any bias. Users were asked to select the best and second best performing method, according to one of two criteria: First, participants were asked to judge each of the 20 videos based on its visual acuity w.r.t. *noise removal*. Second, the users were asked to rate the same 20 sequences for each method w.r.t. *temporal consistency*. The exact description of these two terms was given in the task sheet, see supplementary material. The user study was performed for two groups of methods - academic and commercial. In the first run, our method was compared to two state-of-the-art academic denoising methods; MF2F [10] and UDVD [29]. In the second run, our method was compared to two commercial high-end denoising methods; NeatVideo [1] and DarkEnergy [15]. Figure 6 shows that our denoised videos were the first choice of 70% of the participants among commercial methods and 82% among academic methods. We can further observe that NeatVideo and MF2F are a clear second choice in their respective categories, which is in accordance with the NR-IQA evaluation in Table 2 and visual assessment in Figure 5. Additional results are provided in the supplementary material.

4.4. Ablation Study

We conducted an ablation study regarding the network architecture, where we evaluate the network's performance with respect to the spatial and temporal window size used in our ASwin strategy. Moreover, we investigate the influence of replacing traditional 3D convolutions with ACS convolutions and standard Swin attention with our proposed ASwin attention. The results of all ablation experiments can be

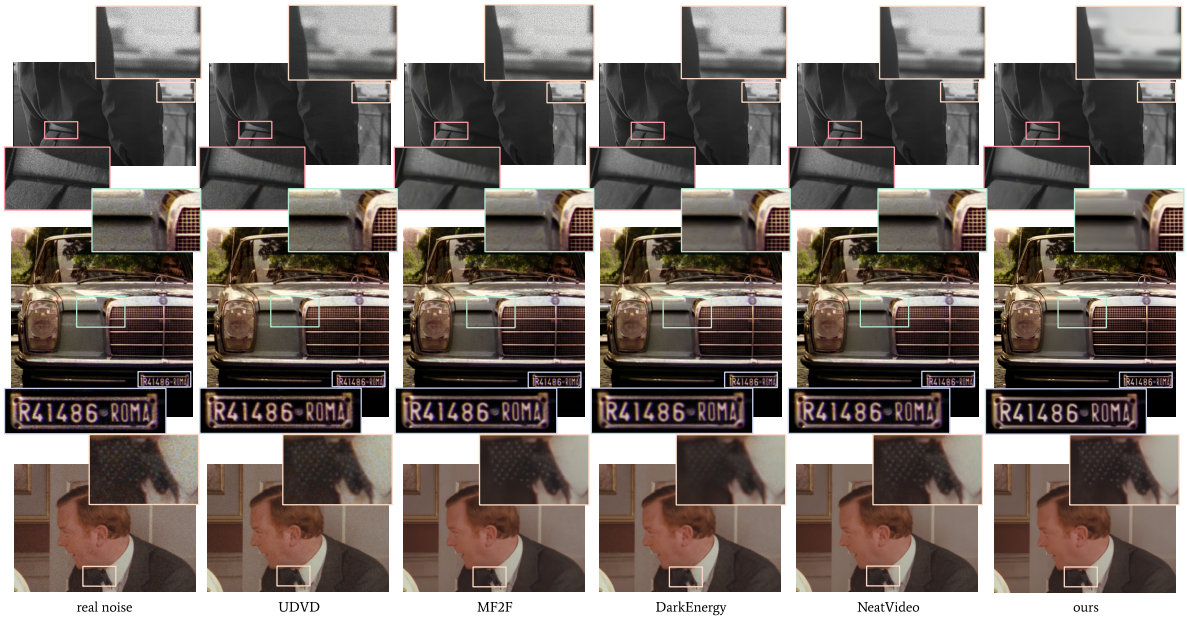
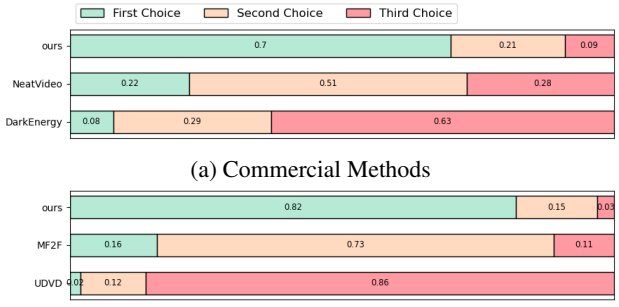


Figure 5: Visualization of the qualitative denoising performance on three digitized analog film scenes.



(a) Commercial Methods

(b) Academic Methods

Figure 6: Combined user study results for noise removal and temporal consistency task. We want to note that the ranking of the evaluated methods stays the same, regardless of the criterion at hand, see supplementary material.

seen in Table 3. It can be observed that increasing the spatial and temporal window size leads to an improved denoising result. The constant memory consumption of ASwin during inference, regardless of the number of considered frames, also allows to increase the spatial window size further than if standard Swin attention would be used. We observed slightly worse results when using standard Swin attention instead of our proposed ASwin attention and also when incorporating full 3D convolutions instead of ACSconv. Due to the increased memory consumption, directly caused by these changes, the architecture setup had to be changed for these experiments, i.e., reducing the training batch size as well as the spatial and temporal size of the attention windows, which explains the slightly worse performance.

spatial ASwin window size			
$h \times w \times t$	***4x4x6***	***8x8x6***	***16x16x6***
PSNR	36.84	36.96	37.12
temporal ASwin window size			
$h \times w \times t$	***16x16x2***	***16x16x3***	***16x16x6***
PSNR	36.86	36.95	37.12
main block configuration			
	Swin+ACSconv	***Swin+3Dconv***	***ASwin+ACSconv***
PSNR	36.92	36.97	37.12

Table 3: Results of ablation experiments.

5. Conclusion

In this work, we introduced a lightweight denoising model that combines efficient ACS convolutions with a novel attention block. The frame-wise aggregation of shifted windows (ASwin) results in a constant memory footprint regardless of the number of considered frames. A comparison on Gaussian video denoising demonstrated that our model yields results close to the state-of-the-art – at only a fraction of the runtime and memory consumption. Moreover, on the challenging task of blind real-world denoising of digitized analog film footage, our model outperforms the state-of-the-art qualitatively and quantitatively as demonstrated in a user study and a non-reference image quality analysis.

Acknowledgement

This work was supported by the FFG-Program BRIDGE with short title RE:Color (No. 877161).

References

[1] ABSoft. Neat video., 2022. https://www.neatvideo.com.

[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, 2018.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. In *NIPS*, 2016.

[4] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T. Barron. Unprocessing images for learned raw denoising. In *CVPR*, pages 11036–11045. Computer Vision Foundation / IEEE, 2019.

[5] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, pages 12299–12310, 2021.

[6] Michele Claus and Jan van Gemert. Videnn: Deep blind video denoising. In *CVPR Workshops*, pages 1843–1852. IEEE, 2019.

[7] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[8] Jingjing Dai, Oscar C Au, Chao Pang, Wen Yang, and Feng Zou. Film grain noise removal and synthesis in video coding. In *ICASSP*, pages 890–893, 2010.

[9] 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, 2019.

[10] Valéry Dewil, Jérémy Anger, Axel Davy, Thibaud Ehret, Gabriele Facciolo, and Pablo Arias. Self-supervised training for blind multi-frame video denoising. In *WACV*, pages 2724–2734, 2021.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

[12] Thibaud Ehret, Axel Davy, Jean-Michel Morel, Gabriele Facciolo, and Pablo Arias. Model-blind video denoising via frame-to-frame training. In *CVPR*, pages 11369–11378, 2019.

[13] Ankit Gupta, Guy Dar, Shaya Goodman, David Ciprut, and Jonathan Berant. Memory-efficient transformers via top-k attention. 2021.

[14] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *ECCV*, pages 1–14. Springer, 2010.

[15] Cinnafilm Inc. Dark energy., 2022. https://cinnafilm.com/product/dark-energy/.

[16] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *ICCV*, 2021.

[17] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *ACCV (4)*, volume 11364 of *Lecture Notes in Computer Science*, pages 123–141. Springer, 2018.

[18] Diederik P. Kingma and Jimmy L. Ba. ADAM: a method for stochastic optimization. In *ICLR*, 2015.

[19] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *NIPS*, volume 32, 2019.

[20] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[21] Stamatios Lefkimmiatis. Universal denoising networks : A novel cnn architecture for image denoising. In *CVPR*. IEEE Computer Society, 2018.

[22] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *International Conference on Machine Learning*, pages 2965–2974, 2018.

[23] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. Vrt: A video restoration transformer. *arXiv preprint arXiv:2201.12288*, 2022.

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *CVPR*, pages 10012–10022, 2021.

[25] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian. Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. *IEEE Transactions on Image Processing*, 21(9):3952–3966, 2012.

[26] Angshul Majumdar. Blind denoising autoencoder. *IEEE Trans. Neural Networks Learn. Syst.*, 30:312–317, 2019.

[27] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *NIPS*, pages 2802–2810, 2016.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*, pages 234–241. Springer International Publishing, 2015.

[29] Dev Yashpal Sheth, Sreyas Mohan, Joshua L Vincent, Ramon Manzorro, Peter A. Crozier, Mitesh M. Khapra, Eero P. Simoncelli, and Carlos Fernandez-Granda. Unsupervised deep video denoising. In *ICCV*, pages 1759–1768, 2021.

[30] Matias Tassano, Julie Delon, and Thomas Veit. Dvdnet: A fast network for deep video denoising. In *ICIP*, pages 1805–1809. IEEE, 2019.

[31] Matias Tassano, Julie Delon, and Thomas Veit. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *CVPR*, pages 1354–1363, 2020.

[32] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *ICCV*, pages 839–846. IEEE, 1998.

[33] Gregory Vaksman, Michael Elad, and Peyman Milanfar. Patch craft: Video denoising by deep modeling and patch matching. In *CVPR*, pages 2157–2166, 2021.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[35] Philippe Weinzaepfel, Jérôme Revaud, Zaïd Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, pages 1385–1392. IEEE Computer Society, 2013.

[36] Jacky C. K. Yan and Dimitrios Hatzinakos. Signal-dependent film grain noise removal and generation based on higher-order statistics. In *Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics*, pages 77–81. IEEE, 1997.

[37] Jiancheng Yang, Xiaoyang Huang, Yi He, Jingwei Xu, Canqian Yang, Guozheng Xu, and Bingbing Ni. Reinventing 2d convolutions for 3d images. *IEEE J. Biomed. Health Inform.*, 25(8):3009–3018, 2021.

[38] Songhyun Yu, Bumjun Park, Junwoo Park, and Jechang Jeong. Joint learning of blind video denoising and optical flow estimation. In *CVPR workshops*, pages 500–501, 2020.

[39] Zongsheng Yue, Hongwei Yong, Qian Zhao, Deyu Meng, and Lei Zhang. Variational denoising network: Toward blind noise modeling and removal. In *NeurIPS*, pages 1688–1699, 2019.

[40] C Zach, T Pock, and H Bischof. A duality based approach for realtime tv-l1 optical flow. In *DAGM German Conference on Pattern Recognition*, pages 214–223, 2007.

[41] Martin Zach and Erich Kobler. Real-world video restoration using noise2noise. In *Joint Austrian Computer Vision and Robotics Workshop*, pages 145–150, 2020.

[42] Kai Zhang, Yawei Li, Jingyun Liang, Jiezhang Cao, Yulun Zhang, Hao Tang, Radu Timofte, and Luc Van Gool. Practical blind denoising via swin-conv-unet and data synthesis. *arXiv preprint arXiv:2203.13278*, 2022.

[43] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *CoRR*, abs/1608.03981, 2016.

[44] Maria Zontak and Michal Irani. Internal statistics of a single natural image. In *CVPR 2011*, pages 977–984, 2011.