A Hybrid Wavelet and Temporal Fusion Algorithm for Film and Video Denoising

Hannes Fassold, Peter Schallauer
JOANNEUM RESEARCH, DIGITAL – Institute for Information and Communication Technologies
Steyrergasse 17, 8010 Graz, Austria
{hannes.fassold, peter.schallauer}@joanneum.at

Abstract

Noise is an impairment which often occurs in both film and digital video and severely degrades the viewing experience of the content. In this work, we propose a two-phase algorithm for film an video denoising. In the first phase, the concept of semi-local shrinkage functions is used to effectively separate noise from image structure. In the second phase, we show how to fuse the result images of the first phase in a way which is robust to motion-compensation errors. A quantitative evaluation of the proposed algorithm on a realistic test dataset with noise of different coarseness and magnitude shows that the proposed method delivers better results than the video denoising method CVBM3D.

1 Introduction

Noise is an impairment which often occurs in both film and digital video. In film, it is known as film grain and results from the visual accumulation of the silver particles in the film emulsion layers. Digital noise is an inherent characteristic of digital sensors and is due to random fluctuations in the number of received photons. The occurrence of noise, either as film grain or digital noise, lowers the viewing experience considerably. Furthermore, it lowers the compression ratio when encoding the noisy content with commonly used video formats (e.g., MPEG-4 AVC [10]) for storage or delivery, as the high-frequency noise components require more coefficients for encoding. Therefore, denoising of film and video content is an essential step in video processing systems.

One can classify the State-of-the-Art algorithms for film an video denoising roughly into three classes: Patch-based, transform-based and combined algorithms. *Patch-based algorithms* take usage of the temporal and spatial redundancy. Usually, for a small patch in an image, one can find several similar patches in the same image and in temporally neighboring images. So these approaches apply some sort of similarity-weighted averaging over all similar blocks in order to retrieve the denoised block. Algorithms belonging to this class are the Non-Local Means algorithm [2] and regularized variants like [9]. A disadvantage of these methods is that their runtime is high due to the expensive block-matching and the similarity computations and they may smooth low-contrast image areas too much. *Transform-based algorithms* work by transforming the image or a whole spatio-temporal volume into a different base, where the base is chosen so that the image information is concentrated in few coefficients with a high magnitude. Many different transforms have been proposed in the literature like DCT [4], wavelet transforms [6] or the curvelet transform [8]. After transform, different kinds of shrinkage are employed in order to keep to the significant coefficients and finally the inverse transform is applied. Transform based algorithms are attractive because they are fast and have a good noise suppression ability if the threshold is properly chosen. On the other side, they are often prone to generating high-frequency artifacts in the denoised image, especially when non-redundant wavelet transforms are used. *Combined algorithms* incorporate principles from both patch-based and transform-based methods. One of the most famous algorithms of this class is the V-BM3D algorithm [3] and its extension CVBM3D for color video. Its basic principle is to group patches according to their similarity, transform this group via a 3-D wavelet transform and filter it via a shrinkage method. The CVBM3D algorithm is considered as one of the best video denoise algorithms, but due to the complex approach its runtime is very high.

The paper is organized as follows. In section 2 we present the proposed algorithm for film and video denoising and will describe both phases of the algorithm in detail. In section 3 we give an evaluation of the algorithm on a test dataset where realistic noise of different type and magnitude has been added, and section 4 concludes the paper.

2 Proposed algorithm

The proposed hybrid wavelet and temporal fusion algorithm (termed HWTF in the following) consists of two phases. In the first phase, the *hybrid wavelet denoising* (see section 2.1) a wavelet-based denoising is applied using the novel concept of semi-local shrinkage functions. In the second phase, the *robust temporal fusion* (see section 2.2), the result images of the first phase are fused within a certain temporal sliding window in a robust way. In both phases, the neighbor images within the temporal sliding window are motion-compensated, by first calculating the motion field between a respective neighbor image and the center image with the GPU-accelerated variational TV-L1 optical flow algorithm from [12] followed by warping of the neighbor image with the motion field. In the following, each phase will be explained more in detail.

2.1 Hybrid wavelet denoising

The hybrid wavelet denoising algorithm is a novel spatio-temporal transform-based approach which exploits efficiently the inherent spatial and temporal correlations present in typical video content. The term *hybrid* is because as it has attributes of both 2-D (spatial)

and 3-D (spatio-temporal) methods. For the denoising of a certain image, it considers a fixed temporal sliding window of motion-compensated neighbor images around this image. E.g., for a sliding window of size three the previous, current and next image is taken into account. Firstly, each image of the sliding window is motion-compensated with respect to the center image and its wavelet transform is calculated. We employ the stationary wavelet transform [5] which is, due to its redundancy, better suited for denoising than the discrete wavelet transform. Then a so-called *semi-local* shrinkage function is applied on all wavelet components of the center image. The novel concept of semi-local shrinkage functions and their advantages over shrinkage functions operating point-wise are explained in section 2.1.1. Afterwards, the inverse wavelet transform is applied to the denoised center image, yielding the result of the first phase.

2.1.1 Semi-local shrinkage functions

A good shrinkage function should shrink wavelet coefficients representing noise towards zero and keep wavelet coefficients representing image structure. Conventional point-wise shrinkage functions only take into account the wavelet coefficient itself (without its spatial and/or temporal neighborhood) which does not allow a good separation of image structure and noise and leads to high-frequency artifacts in otherwise smooth image regions. Therefore, we introduce the novel concept of *semi-local* shrinkage functions which we define as follows. Let w be a coefficient at a certain (x,y) position of a wavelet component and w_{sm} be the corresponding coefficient from a *smoothed* version of the wavelet component (by spatial and/or temporal smoothing). Then, the generalized formula for a semi-local shrinkage function is defined as

$$v = \frac{\varphi\left(|w_{sm}|\right)}{\epsilon + |w_{sm}|} w$$

where the function $\varphi : \mathbb{R}_0^+ \mapsto \mathbb{R}_0^+$ denotes a certain shrinkage function, ϵ is a small positive number which makes the denominator always positive and v is the result (shrinked) wavelet coefficient. The function φ must be monotonously increasing and has to fullfill $\varphi(t) \leq t$ for all $t \in \mathbb{R}_0^+$. Except for these two constraints it can be designed freely. Note that all commonly employed shrinkage functions like soft, hard or firm shrinkage (see [1]) can be easily transferred into this framework. E.g., for soft shrinkage the equivalent function is

$$\varphi_{soft}(t) = max(t - \lambda, 0)$$

where λ is a non-negative constant. For the calculation of the smoothed wavelet component, we apply a combined spatio-temporal smoothing procedure as follows. Firstly, a *temporal* smoothing is applied by averaging all corresponding wavelet components (e.g., all LH components of level 1) within the whole sliding window. Afterwards, a *spatial* smoothing is applied on the temporally smoothed wavelet component. For the spatial smoothing, preferably an edge-preserving smoothing method (e.g, bilateral filtering [11]) should be employed.

Due the usage of the spatio-temporally smoothed wavelet component w_{sm} a much stronger shrinkage for wavelet coefficients representing noise can be achieved. At the same time, wavelet coefficients which represent image structure are not impacted negatively. Furthermore, the amount of annoying high-frequency artifacts is also reduced significantly.

2.2 Robust temporal fusion

As in the first phase, a certain temporal sliding window of motion-compensated images (here we take the result images of phase one) is used. For calculating the final denoised center image, the task is now to fuse all motion-compensated images of the sliding window in a way that is robust against motion-compensation errors which often appear in fast-motion areas and in occluded areas, even when a high-quality optical flow method is used. In the following, for a certain pixel $\mathbf{x}$, let v_r be its intensity in the center image (either a scalar or vector) and v_k be the corresponding intensity in the neighbor image with index k, $k = 1...m$, where m is the size of the temporal sliding window. For the temporal fusion at a certain pixel $\mathbf{x}$, we propose to employ a similarity-weighted average of all intensities v_k, where the weight for a certain intensity v_k is calculated from its similarity to the center image intensity v_r. This concept is related to the classical spatial neighborhood filters like the Yaruslavsky filter [2] and the bilateral filter [11], but with *temporal* instead of spatial extent. In the following, we formalize the novel generalized framework we developed for the robust temporal fusion. Let $\tau : \mathbb{R}_0^+ \mapsto \mathbb{R}_0^+$ be a monotonously descending function which basically specifies how the intensity deviation to the reference intensity v_r is mapped to a weighting factor. E.g., reciprocal- or exponential-like functions of the form

$$\tau_{div}(t) = \frac{1}{(t + \alpha)^{\omega}}$$

$$\tau_{exp}(t) = e^{-\alpha t^{\omega}}$$

can be used, where α and ω are positive constants. We construct now a function η which maps a neighbor intensity into a weighting factor, in a way which allows for some tolerance δ in the intensity deviation as

$$\eta(v_k) = \tau\left(max\left(|v_k - v_r| - \delta, 0\right)\right)$$

By this construction with a non-negative tolerance constant δ, we assure that minor intensity differences (e.g., due to flicker and noise) do not decrease the weighting factor $\eta(v_k)$. On the other side, all intensity differences bigger than δ will decrease the weighting factor as they indicate a motion-compensation error. The actually used function τ determines how fast the decay of the weight is. If the values v_k are vectors (which is the case for color images), one has to adapt the formula for $\eta(v_k)$ slightly by replacing the absolute value function with some suitable vector norm. Finally, the result v_{tf} of the robust temporal fusion is calculated as the weighted average of all intensities v_k via the formula

$$v_{tf} = \frac{1}{\sum_{k=1}^{m} \eta(v_k)} \sum_{k=1}^{m} \eta(v_k) v_k$$

Table 1. PSNR results for the five sequences of the test dataset. In each row, the upper number gives the PSNR of the proposed HWTF algorithm, and the lower number gives the PSNR of the CVBM3D algorithm. The higher of the two PSNR values is marked bold.

Sequence	Variant 1	Variant 2	Variant 3	Variant 4	Variant 5	Variant 6
clinicip	**43.60**	**42.52**	**40.73**	43.69	**40.61**	**38.09**
	43.50	41.70	39.90	**43.73**	39.53	37.25
flowers	44.12	**43.38**	**42.27**	44.20	**42.13**	**40.08**
	44.28	43.01	41.49	**44.43**	41.31	39.38
paprica	**46.01**	**44.23**	**42.23**	**46.04**	**41.95**	**39.18**
	44.54	42.64	40.66	44.69	40.02	37.36
earth	**46.01**	**43.64**	41.45	**46.15**	**41.26**	38.46
	45.25	43.34	**41.79**	45.39	41.21	**39.15**
cooking	**46.69**	**44.41**	**42.21**	**46.71**	**41.79**	**38.87**
	44.76	42.92	41.23	44.76	40.64	38.29

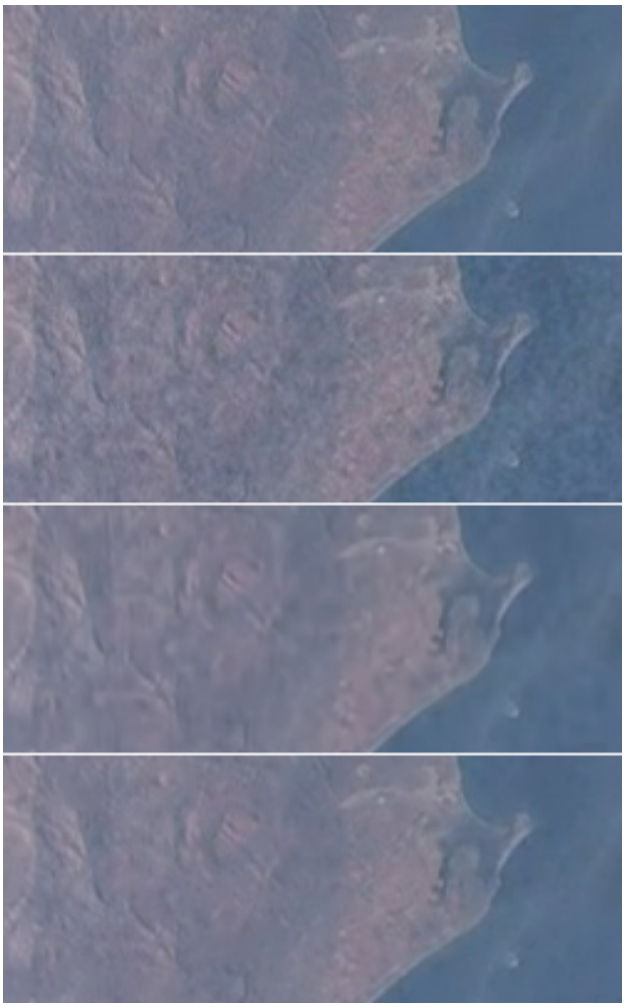

Figure 1. From top to bottom: A region of the original image, the region with noise added (coarse noise, medium magnitude), result of CVBM3D algorithm, result of the proposed HWTF algorithm.

3 Evaluation

We evaluated the proposed video denoising algorithm on a suitable dataset. Commonly used test sequences (foreman, flowergarden, ...) have an unrealistic small resolution compared to recent video cameras which usually have at least HD resolution. Furthermore the usual process of adding Gaussian noise of a certain standard deviation does not reflect the real appearance of film grain noise or digital camera sensor noise which is signal-dependent, spatially correlated and not purely Gaussian. So we generated a dataset of five clean (no apparent noise) video clips with different content characteristics (varying amount of texture and varying levels of motion) and with a resolution of 1280 x 720 pixel. In order to make the addition of noise/grain as natural as possible, we take a noise/grain template, captured from homogenous regions of a real noisy image, and use the texture synthesis method from [7] to create noise with the same appearance as the noise/grain template. We use two different templates, the first one containing fine electronic noise and the second containing coarse film grain. For each noise template and for each clip, we use the method from [7] to add signal-dependent noise in three different magnitudes (low/medium/high). So finally we have 7 variants from each video clip: The clean reference clip, 3 variants with fine noise added (Variant 1 - Variant 3 in Table 1) and 3 variants with coarse noise added (Variant 4 - Variant 6 in Table 1). As the quantitative measure of the denoising quality, we employ the PSNR value. We compare the proposed HWTF algorithm against the CVBM3D algorithm, which is an extension of the V-BM3D algorithm [3] for color video. It is considered currently as the best video denoising algorithm reported in the literature. Both algorithms are run in several parametrizations and for each algorithm the parametrization which gives the highest PSNR is taken. The sliding window size is set to three images for both phases of the proposed HWTF algorithm, whereas the CVBM3D algorithm uses nine images.

In Table 1, the PSNR values are given for the proposed HWTF algorithm (upper numbers) and for the CVBM3D algorithm (lower numbers). One can see that the HWTF algorithm achieves a higher PSNR for most variants of the five test sequences, for some variants of the sequences 'paprica' and 'cooking' it is even almost 2 dB higher. A visual comparison of the algorithm results on the 'earth' sequence of the test dataset is given in Figure 1. One can see that the HWTF algorithm is able to retain fine image structures significantly better. In Figure 2, the algorithm result is

Figure 2. Top: A region of the noisy image, bottom: result of the proposed HWTF algorithm. Best viewed electronically.

shown on a region of an image from the 4K 'coastguard' video provided by Elemental Technologies[1]. The video was recorded using a RED® Epic 4K digital cinema camera in ProRes format. One can see that digital noise has been effectively suppressed, while image details have been preserved.
Regarding runtime, the HWTF algorithm processes one frame in 0.7 seconds, whereas the CVBM3D algorithm (Matlab implemention using C Mex library) needs 5 seconds for one frame. The significant runtime difference may be attributed to the lower computational complexity of the HWTF algorithm, as it employs a significantly smaller temporal sliding window and has no expensive collaborative filtering step.

4 Conclusion

In this paper we proposed a two-phase algorithm for film an video denoising. We employ the concept of semi-local shrinkage function in the first phase, followed by robust temporal fusion in the second phase. A quantitative evaluation of the proposed algorithm shows that the method delivers better results than the state-of-the-art video denoising method CVBM3D, additionally its runtime is significantly lower.

[1] `http://www.elementaltechnologies.com/resources/4k-test-sequences`

5 Acknowledgments

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 600827, DAVID ("Digital AV Media Damage Prevention and Repair"). Furthermore we want to thank Elemental Technologies for providing the 4K test sequence 'coastguard' under the Creative Commons License (`http://creativecommons.org/licenses/by-nd/3.0/deed.en_US`) and NASA for making the 'earth' video publically available.

References

[1] A. Antoniadis. Wavelet methods in statistics: Some recent developments and their applications. *Statistics Surveys*, 1:16–55, 2007.

[2] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, June 2005.

[3] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3d transform-domain collaborative filtering. In *Proc. 15th European Signal Processing Conference (EUSIPCO)*, Poznan, Poland, 2007.

[4] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images. *Image Processing, IEEE Transactions on*, 16(5), May 2007.

[5] J. Fowler. The redundant discrete wavelet transform and additive noise. *Signal Processing Letters, IEEE*, 12(9), Sept 2005.

[6] J. Portilla and E. Simoncelli. Image restoration using gaussian scale mixtures in the wavelet domain. In *Proc.International Conference on Image Processing (ICIP)*, volume 2, Sept 2003.

[7] P. Schallauer and R. Mörzinger. Film grain synthesis and its application to re-graining. In *Proc. SPIE Image Quality and System Performance III*, San Jose, USA, Januar 2006.

[8] J.-L. Starck, E. Candes, and D. Donoho. The curvelet transform for image denoising. *Image Processing, IEEE Transactions on*, 11(6), Jun 2002.

[9] C. Sutour, C.-A. Deledalle, and J.-F. Aujol. Adaptive regularization of the nl-means: Application to image and video denoising. *Image Processing, IEEE Transactions on*, 23, Aug 2014.

[10] A. Tamhankar and K. Rao. An overview of h.264/mpeg-4 part 10. In *Video/Image Processing and Multimedia Communications, 2003. 4th EURASIP Conference focused on*, July 2003.

[11] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, Washington, DC, USA, 1998. IEEE Computer Society.

[12] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic huber-l1 optical flow. In *Proc. of the British Machine Vision Conference (BMVC)*, London, UK, September 2009.