DUAL DOMAIN VIDEO DENOISING WITH OPTICAL FLOW ESTIMATION

A. Buades and J. L. Lisani

Universitat Illes Balears
Spain
toni.buades, joseluis.lisani@uib.es

ABSTRACT

We propose a new video denoising algorithm combining state of the art image and video denoising algorithms. We extend the DDID [1] algorithm to video sequences and then combine it with the SPTWO [2] method. The experimentation illustrates how the new method keeps the best of each algorithm, being superior both visually and numerically to other state of the art techniques.

***Index Terms*—** video denoising; optical flow; Fourier shrinkage; dual domain

1. INTRODUCTION

Techniques for noise removal in digital images comprise transform thresholding, local averaging, patch based methods and variational techniques. Nowadays, state of the art methods actually combine two or three of these techniques.

BM3D [3] combined patch based grouping and thresholding methods, using a 3D DCT transform. Several algorithms appeared combining the grouping of similar patches and the learning of an adapted basis via PCA or SVD decomposition [4, 5]. In [6] a Gaussian model is fitted for each group of similar patches while in [7] the shape of the patch is adapted to the image before computing a PCA description.

The Dual-Domain Image Denoising (DDID), was proposed by Knaus et al. in [1], and combines the bilateral filter [8] and classical attenuation of Fourier coefficients [9, 10, 11, 12]. The ringing artifacts introduced by the Fourier coefficients attenuation are reduced by using spatially uniform regions computed from a guide image. This guide image was the result of a previous iteration of the algorithm, but soon several authors [13, 14, 15] proposed to replace it by a clean image obtained with some other denoising method (e.g. BM3D or NL-Bayes).

Patch based methods as NL-means [16] or BM3D [7] and NL-Bayes [6] can be easily adapted to video just by extending the neighboring area to the adjacent frames. The performance of these methods can be improved by taking into account the non static nature of videos. Kervrann and Boulanger [17] extended the NL-means to video by growing adaptively the spatio-temporal neighborhood. VBM4D [18], exploits the mutual similarity between 3D spatio-temporal volumes constructed by tracking blocks along trajectories defined by the motion vectors.

In [2] the authors proposed to combine optical flow estimation and patch based methods for video denoising. The algorithm, called SPTWO, first compensates the motion building a static 3D volume. Then, spatiotemporal 3D windows are used for selecting groups of similar patches which are denoised by an adapted PCA transform.

In this paper we propose to adapt the DDID algorithm to video sequences and then to combine it with the SPTWO method. The paper is organized as follows: Section 2 summarizes the method proposed in [2] for white noise removal. Section 3 describes DDID. The new denoising algorithm is described in Section 4 and experimental results are shown in Section 5. Finally some conclusions are presented in Section 6.

2. IMAGE SEQUENCE WHITE NOISE REMOVAL

In this section, we briefly review the image sequence denoising algorithm proposed in [2]. We describe the complete algorithm for denoising a frame $\boldsymbol{y}_k$ from a sequence $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N\}$. The same procedure is applied sequentially to denoise all the frames of the sequence.

First, the optical flow between $\boldsymbol{y}_k$ and adjacent frames in a temporal neighborhood is computed and used for warping these frames onto $\boldsymbol{y}_k$. The algorithm uses a 3D volumetric approach to search for similar patches, while still 2D image patches are used for denoising. For each patch P of the reference frame $\boldsymbol{y}_k$, the patch $\mathcal{P}$ referring to its extension to the temporal dimension is considered, having M times more pixels than the original one (assuming M patches in the temporal neighborhood). Since the images have been resampled according to the estimated flow, the data is supposed to be static. The algorithm looks for the K extended patches closest to $\mathcal{P}$. As each extended patch contains M 2D image patches, the group contains $K \cdot M$ selected patches. The Principal Component Analysis (PCA) of these patches is computed and their

The authors were supported by grant TIN2014-53772. The second author was also supported by grant TIN2014-58662-R.

denoised counterparts are obtained by thresholding of the coefficients. As proposed in [4], the decision of canceling a coefficient is not taken depending on its magnitude, but on the magnitude of the associated principal value. The whole patch is restored in order to obtain the final estimate by aggregation.

A second iteration of the algorithm is performed using the "oracle" strategy. Once the whole sequence has been restored, the algorithm is re-applied on the initial noisy sequence, but motion estimation and patch selection are performed on the result of the first iteration. Analogously, the PCA is computed in the set of already denoised patches while the coefficients of noisy patches in the computed basis are modified by a Wiener filter strategy.

Color images are denoised directly without the use of any color decorrelating transform. Each color patch is considered as a vector with three times more components than in the single channel case.

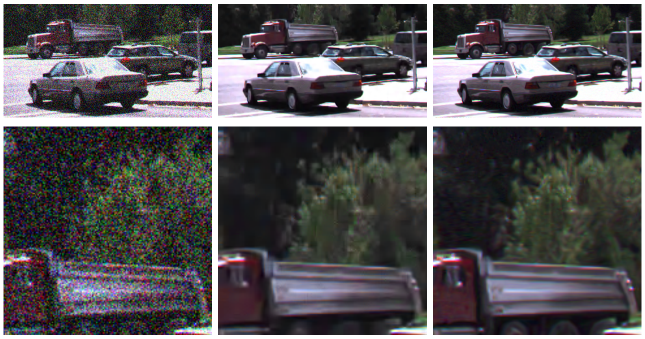

Fig. 1: Comparison of results on white noise removal between SPTWO [2] and VBM3D [19]. First row, from left to right: noisy central frame of the sequence (standard deviation 50), denoising results with VBM3D and SPTWO. Second row: details. RMSE errors: VBM3D 9.92, SPTWO 8.91.

Figure 1 displays the noisy and denoised central frames for a sequence using the SPTWO [2] and VBM3D [19] denoising algorithms. Both algorithms are able to remove the noise but SPTWO [2] better preserves details and texture.

3. DUAL DOMAIN IMAGE DENOISING

This section describes the DDID algorithm [1] using the alternative interpretation proposed in [14]. To denoise a pixel p from a noisy image, DDID extracts a pixel block $\mathcal{N}_p$ around it (denoted y) and the corresponding block g from the guide image. The blocks are processed, using a weight function k, to eliminate discontinuities that may cause artifacts in the subsequent frequency-domain denoising. The weight function k is derived from g and has the form of the bilateral filter [20, 8]

$$k(q) = \exp\left(-\frac{|g(q)-g(p)|^2}{\gamma_r \sigma^2}\right) \exp\left(-\frac{|q-p|^2}{2\sigma_s^2}\right). \quad (1)$$

The first term identifies the pixels with a similar color to p in the guide, while the second term imposes a smooth shape of the filter in the spatial domain. The parameters σ_s and γ_r are specific of the algorithm, and σ is the standard deviation of the noise. The filter is applied on y and g in order to remove their discontinuities yielding y_m and g_m,

$$y_m(q) = k(q)y(q) + (1-k(q))\tilde{s}, \quad (2)$$

$$g_m(q) = k(q)g(q) + (1-k(q))\tilde{g}, \quad (3)$$

where $q \in \mathcal{N}_p$ and $\tilde{s}$ and $\tilde{g}$ are the average, weighted by k, of the pixels in $\mathcal{N}_p$. In this way, parts of the block similar to the central pixel are kept in the filtered block while the average value is assigned to the rest.

The modified block y_m is denoised by shrinkage of its Fourier coefficients using the following shrinkage factor

$$W(f) = \begin{cases} 1 & \text{if } f = 0 \\ \exp\left(-\frac{\gamma_f \sigma_f^2}{|G(f)|^2}\right) & \text{otherwise,} \end{cases} \quad (4)$$

where $S(f)$ and $G(f)$ are, respectively, the Fourier transforms of y_m and g_m (which acts as an oracle), γ_f is a parameter of the algorithm and σ_f^2 is the variance of the noise at frequency f, computed as $\sigma_f^2 = \sigma^2 \sum_{q \in \mathcal{N}_p} k(q)^2$.

Since discontinuities have been removed from the blocks, filtering in the Fourier domain doesn't introduce ringing. Finally, the denoised value of the central pixel is recovered by reversing the Fourier transform. This process is repeated for every pixel of the image. For color images, k is computed using the Euclidean distance on the YUV color space, and the shrinkage is done independently on each channel. The process is described in Algorithm 1.

Algorithm 1 DDID

```
Input: Noisy image $\boldsymbol{y}$, partially denoised image $\boldsymbol{g}$, noise variance $\sigma^2$
Output: Denoised image $\boldsymbol{y}_{denoised}$
for all pixels $p \in \boldsymbol{y}$ do
    $y, g \leftarrow$ ExtractPatches$(\boldsymbol{y}, \boldsymbol{g}, p)$
    $k \leftarrow$ ComputeFilter$(g, p)$        //Equation (1)
    $y_m, g_m \leftarrow$ FilterPatches$(y, g, k, p)$//Equations (2) and (3)
    $S(f) \leftarrow FFT(y_m), G(f) \leftarrow FFT(g_m)$
    $W(f) \leftarrow$ Apply Equation (4) to $G(f)$
    $\boldsymbol{y}_{denoised}(p) \leftarrow$ IFFT$\{S(f) \cdot W(f)\}(p)$
end for
```

4. PROPOSED ALGORITHM

In this section we propose an adaptation of the DDID algorithm to denoise video sequences. The new algorithm, which we call DDVD, is described in Section 4.1. We then combine this extension with the state-of-the art SPTWO method (described in Section 2).

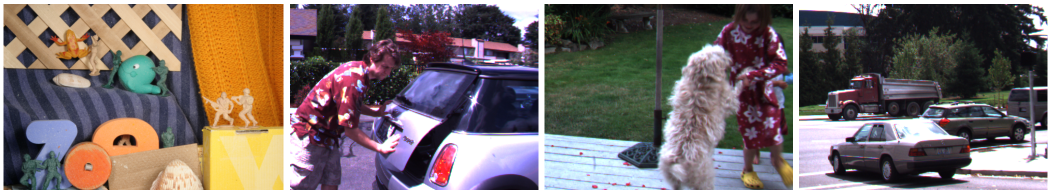

Fig. 2: Central frame of the original color sequences used in our tests. From left to right and from top to bottom: army, cooper, dog, truck. All the sequences are composed of 8 frames.

The final algorithm, which we call DDVDO, is described in Algorithm 3. The first stage of the SPTWO algorithm is used as guide for the new introduced video DDID. Then, the result of DDVD acts as an oracle for the second iteration of SPTWO.

4.1. Dual Domain Denoising of Videos (DDVD)

The adaptation of DDID to video sequences is described in Algorithm 2. The algorithm takes as input a noisy sequence $\boldsymbol{y}$ and a guide sequence $\boldsymbol{g}$ composed by N frames $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N\}$ (resp. $\{\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_N\}$). We describe how to denoise a particular frame bmg_i. Repeating the method taking each frame as reference the whole sequence is denoised.

First, the optical flow between $\boldsymbol{g}_i$ and adjacent frames in a temporal neighborhood $\mathcal{T}_i$ (of size M) is computed and used for warping these frames onto $\boldsymbol{g}_i$. The same flow is used to warp the frames in a temporal neighborhood of $\boldsymbol{y}_i$ onto this frame. Since the images have been resampled according to the estimated flow, the warped sequences (denoted S_i^g and S_i^y, respectively) are supposed to be static. However, inaccuracies and errors in the computed flow and the presence of occlusions may introduce artifacts. Occluded pixels are detected by checking the forward-backward coherence of the computed flow. Since for the occluded pixels the subsequent steps may introduce artifacts in the denoising result, we opt for assigning to them the value of the guide sequence.

For each non occluded pixel p the sets of patches centered at p for both the noisy and guide warped sequences are extracted. Denote them as $\mathcal{P}_p^y$ and $\mathcal{P}_p^g$, respectively. A unique weight function k for each patch in $\mathcal{P}_p^y$ and $\mathcal{P}_p^g$ is defined as

$$k(q) = \exp\left(-\frac{\sum_{j\in\mathcal{T}_i} |g_i(q) - g_j(p)|^2}{\gamma_r M \sigma^2}\right) \exp\left(-\frac{|q-p|^2}{2\sigma_s^2}\right). \tag{5}$$

Remark that color differences in Equation (1) have been replaced by the average of these differences along the stabilized sequence. That is, the same kernel k is used for all the patches in the static 3D neighborhood.

Each pair of corresponding (i.e. belonging to the same frame) patches y and g from $\mathcal{P}_p^y$ and $\mathcal{P}_p^g$ are filtered using Equations (2), (3) and (5), obtaining y_m and g_m. The sets of modified patches form two 3D structures, $\mathcal{P}_p^{y_m}$ and $\mathcal{P}_p^{g_m}$, associated to the original noisy and guide sequences. As with DDID, the next step is to apply a shrinkage of the Fourier domain coefficients of the filtered noisy input using as an oracle the filtered guide input. Instead of using the Fourier transform of a 2D patch, a 3D transform is applied to the warped static data. For practical reasons, the 3D Fourier transforms are decomposed in a 2D transform in the spatial domain followed

Algorithm 2 DDVD

Input: Noisy video sequence $\boldsymbol{y}$, partially denoised video sequence $\boldsymbol{g}$, noise variance σ^2
Output: Denoised video sequence $\boldsymbol{y}_{denoised}$

```
for each frame $\boldsymbol{y}_i$ and $\boldsymbol{g}_i$ in input sequences do
    Build static sequence $S_i^y$ (resp. $S_i^g$) around $\boldsymbol{y}_i$ (resp. $\boldsymbol{g}_i$):
        $\mathcal{T}_i$=temporal neighborhood ($M$ adjacent frames)
        Compute optical flow from $\boldsymbol{g}_i$ to all $\boldsymbol{g}_j \in \mathcal{T}_i$.
        Warp all $\boldsymbol{g}_j$ and $\boldsymbol{y}_j \in \mathcal{T}_i$ using this flow.
    for each pixel $p$ do
        if $p$ is not occuded then
            $\mathcal{P}_p^y$=patches centered at $p$ for all frames in $S_i^y$.
            $\mathcal{P}_p^g$=patches centered at $p$ for all frames in $S_i^g$.
            $k \leftarrow$ ComputeFilter($\mathcal{P}_p^g, p$)        //Equation (5)
            $\mathcal{P}_p^{y_m} \leftarrow \emptyset$, $\mathcal{P}_p^{g_m} \leftarrow \emptyset$ (sets of modified patches)
            for each patch $y \in \mathcal{P}_p^y$ do
                $g \leftarrow$ corresponding patch to $y$ in $\mathcal{P}_p^g$
                $y_m, g_m \leftarrow$ FilterPatches($y, g, k, p$)
                $\mathcal{P}_p^{y_m} \leftarrow \mathcal{P}_p^{y_m} \cup y_m$, $\mathcal{P}_p^{g_m} \leftarrow \mathcal{P}_p^{g_m} \cup g_m$
            end for
            $S(f) \leftarrow FFT(\mathcal{P}_p^{y_m})$, $G(f) \leftarrow FFT(\mathcal{P}_p^{g_m})$
            $W(f) \leftarrow$ Apply Equation (4) to $G(f)$
            $\boldsymbol{y}_{i_{denoised}}(p) \leftarrow$ IFFT$\{S(f) \cdot W(f)\}(p)$
        else
            Set $\boldsymbol{y}_{i_{denoised}}(p) \leftarrow \boldsymbol{g}_i(p)$
        end if
    end for
end for
```

Algorithm 3 DDVDO

Input: Noisy video sequence $\boldsymbol{y}$, noise variance σ^2
Output: Denoised video sequence $\boldsymbol{y}_{denoised}$

```
$\boldsymbol{g_1}$=SPTWO($\boldsymbol{y}, \sigma$)
$\boldsymbol{g_2}$=DDVD($\boldsymbol{y}, \boldsymbol{g_1}, \sigma$)
$\boldsymbol{y}_{denoised}$=SPTWO($\boldsymbol{y}, \boldsymbol{g_2}, \sigma$)
```

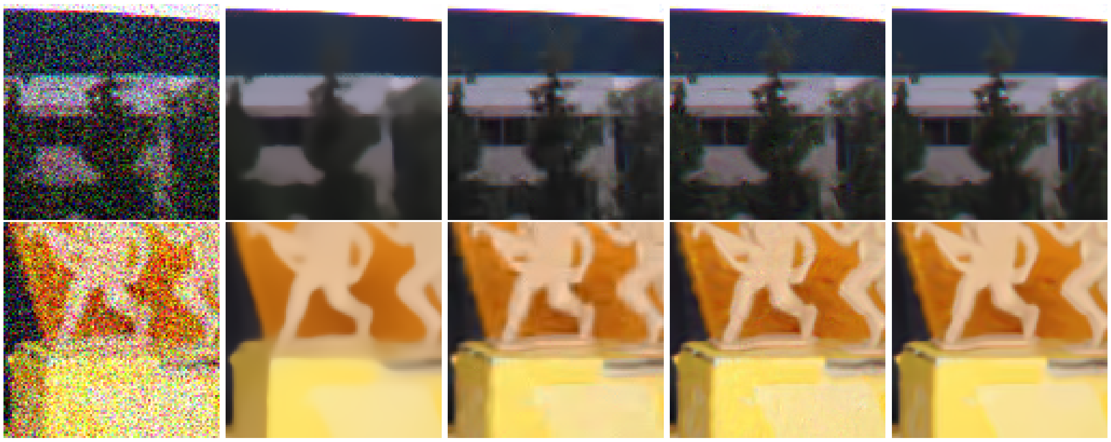

Fig. 3: Detail of the denoised central frame (top $\sigma = 40$, below $\sigma = 50$). From left to right: noisy, NLDD, VBM3D, SPTWO, and DDVDO.

by a 1D transform in the temporal domain. For color images, k is computed using the Euclidean distance on the YUV color space, and the shrinkage is done independently on each channel.

	army	cooper	dog	truck	**average**
$\sigma = 10$					
NLDD	3.74	3.89	4.25	3.87	3.94
VBM3D	2.96	4.31	4.48	3.76	3.88
SPTWO	2.77	**4.06**	**4.06**	**3.61**	**3.62**
DDVDO	**2.72**	4.11	4.09	3.69	3.65
$\sigma = 20$					
NLDD	5.64	6.94	6.75	6.30	6.41
VBM3D	4.42	6.84	6.37	5.70	5.83
SPTWO	3.88	**6.19**	**5.66**	**5.31**	**5.26**
DDVDO	**3.87**	6.20	5.68	5.32	5.27
$\sigma = 30$					
NLDD	6.35	9.55	8.27	8.21	8.10
VBM3D	5.54	8.87	7.75	7.30	7.37
SPTWO	4.62	7.85	**6.79**	6.66	6.48
DDVDO	**4.62**	**7.81**	6.82	**6.62**	**6.47**
$\sigma = 40$					
NLDD	6.79	11.88	9.16	9.84	9.41
VBM3D	6.40	10.49	8.76	8.63	8.57
SPTWO	5.28	9.19	**7.69**	7.81	7.49
DDVDO	**5.25**	**9.13**	7.70	**7.75**	**7.46**
$\sigma = 50$					
NLDD	7.66	14.07	10.53	11.53	10.95
VBM3D	7.34	12.01	9.90	9.92	9.79
SPTWO	5.93	10.33	8.57	8.78	8.40
DDVDO	**5.84**	**10.23**	**8.54**	**8.63**	**8.31**

Table 1: RMSE results for color sequences in Fig. 1. The values correspond to the RMSE (averaged over the three channels) computed for the central frame of each sequence. The average RMSE for each method and each noise level is displayed in the last column.

5. EXPERIMENTATION

In this section we illustrate the performance of the proposed method DDVDO and compare it to state-of-the art algorithms for video denoising VBM3D [19] and SPTWO [2]. We also compare it to NLDD [14], applied on a frame-by-frame basis to the video sequence. Four sequences, composed of 8 frames each, have been used in the tests (see Fig. 2). A white Gaussian noise with increasing levels of variance σ^2 has been added to the sequences, which have been then denoised using the different methods. Table 1 displays the Root Mean Squared Error (RMSE) between the denoised results and the original (noise-free) sequences, computed for the central frame. The values correspond to the average RMSE value of the three channels. Notice that the smaller RMSE values are obtained, in general, with the proposed algorithm, especially for high values of σ.

Fig. 3 illustrates the visual quality of the denoising results of the compared methods. SPTWO [2] and the proposed algorithm DDVDO preserve more texture and details than the rest of the algorithms. Spurious noise artifacts remain in the SPTWO result while these are completely removed by the proposed algorithm.

The experiments were performed on a PC running a Linux operating system (Ubuntu 14.04LTS), with 4 Intel Core i5-4460 CPU at 3.20GHz and 8GB RAM. The average computation time, per frame, for a video sequence with 8 frames of 584 × 388 pixels, was 5 minutes. Each step of SPTWO took 1 min./frame while DDVD took up to 3 min./frame. Remark that the code (in C) has not been optimized. Although some parts of the algorithms benefit from a parallel implementation, the parallel execution of other parts (e.g. the FFT transforms in DDVD, and the warping of the frames in both algorithms) could greatly increase the computation speed.

6. CONCLUSIONS

We have proposed an extension of the DDID [1] to video sequences and combined this algorithm with state of the art SPTWO [2]. The new procedure keeps the properties of texture and detail preservation of SPTWO and the non presence of spurious noise artifacts of DDID. The resulting method has shown to be superior both visually and numerically to state of the art algorithms.

7. REFERENCES

[1] C. Knaus and M. Zwicker, “Dual-domain image denoising,” in *Proceedings of IEEE International Conference on Image Processing*, 2013.

[2] M. Miladinović A. Buades, J.L. Lisani, “Patch-based video denoising with optical flow estimation,” *IEEE Transactions on Image Processing*, vol. 25, no. 6, 2016.

[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3D transform-domain collaborative filtering,” *IEEE Transactions on image processing*, vol. 16, pp. 2007, 2007.

[4] L. Zhang, W. Dong, D. Zhang, and G. Shi, “Two-stage image denoising by principal component analysis with local pixel grouping,” *Pattern Recognition*, vol. 43, no. 4, pp. 1531–1549, 2010.

[5] J. Orchard, M. Ebrahimi, and A. Wong, “Efficient Non-Local-Means Denoising using the SVD,” in *Proceedings of The IEEE International Conference on Image Processing*, 2008.

[6] M. Lebrun, A. Buades, and J.-M. Morel, “A nonlocal bayesian image denoising algorithm,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1665–1688, 2013.

[7] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, et al., “BM3D image denoising with shape-adaptive principal component analysis,” *Proceedings of the Workshop on Signal Processing with Adaptive Sparse Structured Representations, Saint-Malo, France*, April 2009.

[8] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proceedings of IEEE International Conference on Computer Vision*, 1998.

[9] D. L. Donoho J. L. Starck, E. J. Candes, “The curvelet transform for image denoising,” *IEEE Transactions on Image Processing*, vol. 11, no. 6, 2002.

[10] C.Z. Deng H.Q. Li, S.Q. Wang, “New image denoising method based wavelet and curvelet transform,” in *Proceedings of WASE International Conference on Information Engineering*, 2009.

[11] D. Gnanadurai and V. Sadasivam, “Image denoising using double density wavelet transform based adaptive thresholding technique,” *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 3, no. 1, 2005.

[12] Guillermo Sapiro Guoshen Yu, “DCT image denoising: a simple and effective image denoising algorithm,” *Image Processing On Line*, 2011.

[13] C. Knaus and M. Zwicker, “Progressive image denoising,” *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 3114–3125, 2014.

[14] N. Pierazzo, M. Lebrun, M.E. Rais, J.M. Morel, and G.Facciolo, “Non-local dual image denoising,” in *Proceedings of IEEE International Conference on Image Processing*, 2014.

[15] N. Pierazzo, M. Lebrun, M.E. Rais, J.M. Morel, and G.Facciolo, “DA3D: Fast and data adaptive dual domain denoising,” in *Proceedings of IEEE International Conference on Image Processing*, 2015.

[16] A. Buades, B. Coll, and J.M. Morel, “A non local algorithm for image denoising,” *IEEE Computer Vision and Pattern Recognition*, vol. 2, pp. 60–65, 2005.

[17] J. Boulanger, C. Kervrann, and P. Bouthemy, “Space-time adaptation for patch-based image sequence restoration,” *IEEE Trans. PAMI*, vol. 29, no. 6, pp. 1096–1102, 2007.

[18] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, “Video denoising using separable 4D nonlocal spatiotemporal transforms,” in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2011, vol. 787003.

[19] K. Dabov, A. Foi, and K. Egiazarian, “Video denoising by sparse 3D transform-domain collaborative filtering,” in *Proceedings of the 15th European Signal Processing Conference*, 2007, pp. 145–149.

[20] L. P. Yaroslavsky, “Local adaptive image restoration and en- hancement with the use of DFT and DCT in a running window,” in *Proceedings of SPIE*, 1996.