Using the Higher Order Singular Value Decomposition (HOSVD) for Video Denoising

Ajit Rajwade, Anand Rangarajan and Arunava Banerjee

Department of CISE, University of Florida, Gainesville (USA)
{avr,anand,arunava}@cise.ufl.edu

Abstract

We present an algorithm for denoising of videos corrupted by additive i.i.d. zero mean Gaussian noise with a fixed and known standard deviation. Our algorithm is patch-based. Given a patch from a frame in the video, the algorithm collects similar patches from the same and adjacent frames. All the patches in this group are denoised using a transform-based approach that involves hard thresholding of insignificant coefficients. In this paper, the transform chosen is the higher order singular value decomposition of the group of similar patches. This procedure is repeated across the entire video in sliding window fashion. We present results on a well-known database of eight video sequences. The results demonstrate the ability of our method to preserve fine textures. Moreover we demonstrate that our algorithm, which is entirely driven by patch-similarity, can produce mean-squared error results which are comparable to those produced by state of the art techniques such as [5], as also methods such as [11] that explicitly use motion estimation before denoising.

1 Introduction

Video denoising is an important application in the field of computer vision or signal processing. Videos captured by digital cameras are susceptible to corruption by noise from various sources: film grain noise, noise due to insufficient bit-rate during transmission, mechanical damage to the DVD, insufficient lighting during exposure time, and so on. The restoration of such videos can have broad applications in the film industry, the communication of multimedia, in remote sensing, medical imaging, and also for plain aesthetic purposes.

The literature on video denoising contains several methods that use shrinkage of coefficients measured on fixed bases such as various types of wavelets. Examples include the work in [1] and [14]. However, the image denoising community has witnessed rapid advances in methods that *infer* 'optimal' bases for denoising image patches. These bases can be learned offline from a representative set of image patches, though dictionaries are often learned *in situ* from the noisy image itself [9]. Several of these methods learn a single global dictionary to sparsely represent the patches in the image [9], some others cluster similar patches *a priori* and learn a single dictionary for each cluster separately [4], whereas a third category of methods learn pointwise varying bases, i.e. separate bases for a fixed size patch located at each pixel [12], [6]. The method we present in this paper belongs to this third category. Given a 'reference' patch from a frame in the video to be denoised, the data for learning the bases consist of the patches from adjacent video frames that are similar to that patch. Thus, our approach can be included in the paradigm of non-local denoising which has emerged very successful in recent times, beginning with approaches such as NL-means (for image and video denoising) [2] and culminating in state of the art approaches such as block-matching-3D ('BM3D' [6]). The BM3D method treats the group of similar patches as a 3D stack and denoises all the patches *jointly* using a fixed 3D transform. This joint filtering has been demonstrated to be more effective in filtering several fine textures than individual filtering of each patch using a 2D transform [6].

The current literature on video denoising indicates two divergent schools of thought. There exist papers such as [2], [5] which do not perform any motion estimation prior to smoothing the video. Their main

argument is that the well-known aperture problem in optical flow actually *helps* the denoising process. In fact, a video sequence contains many more patches that are similar to a given reference patch (as compared to a single image) and this added redundancy can enhance the video denoising results. On the other hand, there exist papers such as [11], [3] which are proponents of prior motion estimation and correction. In this paper, we choose not to perform motion estimation and perform denoising solely on the basis of non-local patch similarity. We present arguments and some empirically driven reasons for this later in Section 3 where we present experimental results. Our work in this paper is based on ideas from our earlier work [13]. The contribution of this paper is the extension of our earlier idea to video denoising with entirely new experimental results.

This paper is organized as follows. Section 2 reviews the theoretical background of our technique. Several experimental results are presented in Section 3. Our results are compared to video-BM3D, which is considered the state of the art in video denoising. We conclude in Section 4.

2 Theory

Let I_n be the corrupted version of a clean image I_t under the action of noise from $\mathcal{N}(0, \sigma)$. Consider a reference patch P_n in I_n and let its underlying clean patch in I_t be P_t. Suppose we computed K patches $\{Q_{ni}\}$ that were 'similar' to P_n from I_n. The similarity metric is detailed in Section 2.2. The principal components of these patches are given by the eigenvectors U_n of the correlation matrix $C_n = \frac{1}{K}\sum_{i=1}^{K} Q_{ni}Q_{ni}^T$. If such a set of eigenvectors is computed for each patch, we get a set of pointwise varying orthonormal bases. Patches from I_n can be denoised by projecting them onto the orthonormal basis computed for each patch, followed by hard-thresholding or some other method to attenuate the smaller coefficient values (which are assumed to consist of mainly noise). This spatially varying PCA approach is presented in [12]. Now assume an ideal situation where all the K patches $\{Q_{ni}\}$ happened to be noisy versions of P_t. In such a case, we see that as $K \to \infty$, $C_n \to C_t + \sigma^2 Id$, where $C_t \stackrel{\text{def}}{=} P_t P_t^T$ and where Id is the identity matrix. The eigenvectors of C_n then have a very good chance of capturing all the structural information in P_t and hard thresholding the insignificant coefficients of the bases derived in this manner will most likely yield a very good quality denoised output.

However, such a situation is usually not possible in most natural images. In fact, the patches that qualify as 'similar' will usually not be exact copies of P_t modulo noise. Hence, we adopt the following principle to further constrain our solution: if a group of patches are similar to one another in the noisy image, the denoising procedure should take this fact into account and not filter the individual patches from the group independently. Bearing this is in mind, we group together similar patches and represent them in the form of a 3D stack as in Equation 1. The main idea is that the filtering is performed not only across the length and breadth of each individual (2D) patch, but also in the third dimension so as to allow for similarity between intensity values at corresponding pixels of the different patches. The idea of joint filtering of multiple patches has been implemented earlier in the BM3D algorithm [6], but with *fixed* bases such as DCT, Haar or Biorthogonal wavelets. However, in this paper, we use this idea to *learn* spatially adaptive bases, which we choose to be the higher order singular value decomposition (HOSVD) bases of the 3D stack of patches. An example in Figure 1 illustrates the superiority of our HOSVD approach over PCA, for denoising a texture image. The third and fourth row of Figure 1 show the application of coefficient thresholding for smoothing of the 11 structurally similar patches of size 64×64 using the PCA and HOSVD transforms respectively, while the last two rows show the filtered patches after the averaging operations (employing the same criteria for patch similarity and coefficient thresholding). These figures reveal that HOSVD preserves the finer textures on the table-cloth surface much better than PCA which almost erases those textures.

2.1 Implementation of the HOSVD for Video Denoising

Given a $s \times s$ reference patch P_n in the noisy image I_n, we create a stack of K similar patches. Here similarity is defined as in Section 2.2. Let us denote the stack as $\mathcal{F} \in R^{s \times s \times K}$. The HOSVD of this stack given as follows [7]:

$$\mathcal{F} = S \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \tag{1}$$

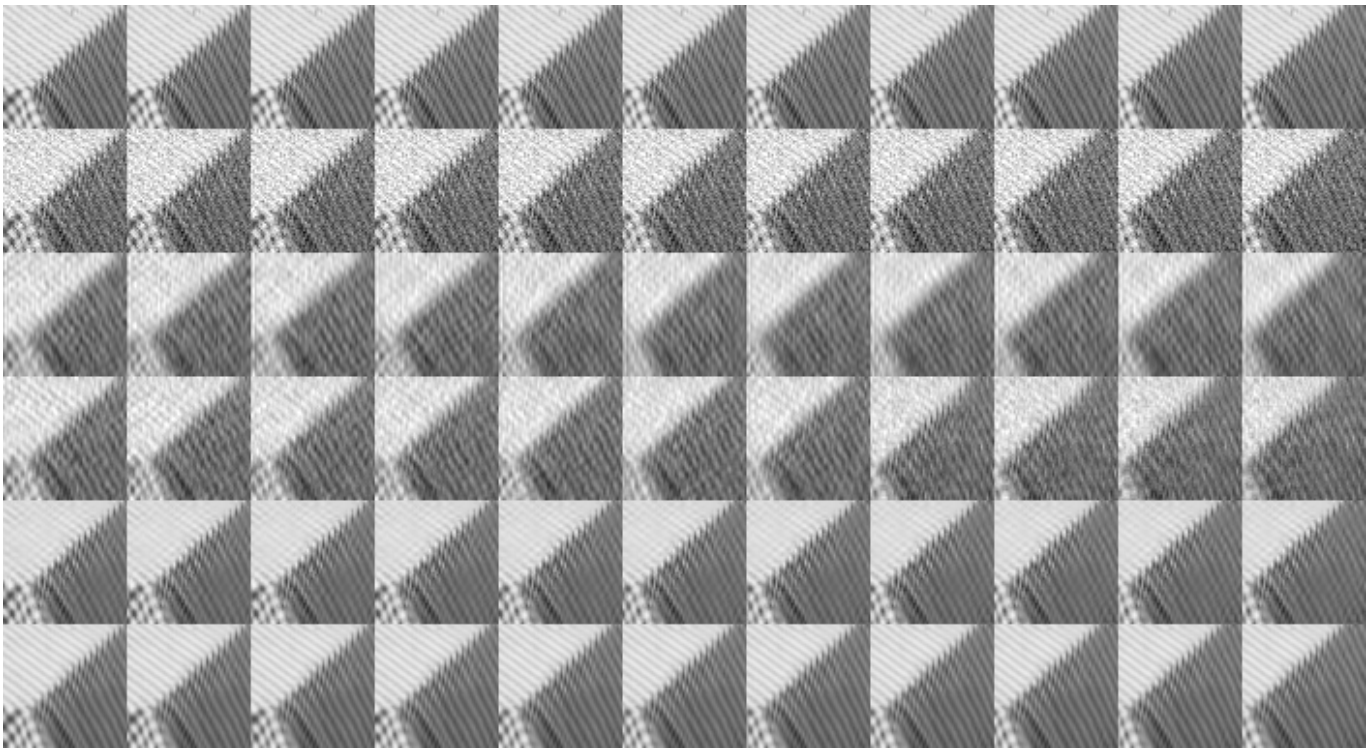

Figure 1: Eleven patches of size 64×64 from a textured portion of the original Barbara image (row 1), its noisy version under $\mathcal{N}(0, 20)$ (row 2), from the PCA output before averaging (row 3), from the HOSVD output before averaging (row 4), from the PCA output after averaging (row 5), from the HOSVD output after averaging (row 6). Zoom into pdf file for a better view.

where $V^{(1)} \in R^{s \times s}$, $V^{(2)} \in R^{s \times s}$ and $U^{(3)} \in R^{K \times K}$ are orthonormal matrices, and S is a 3D coefficient array of size $s \times s \times K$. Here, the symbol $\times_j$ stands for the j^{th} mode tensor product defined in [7]. The orthonormal matrices $V^{(1)}$, $V^{(2)}$ and $V^{(3)}$ are in practice computed from the SVD of the unfoldings $\mathcal{F}_{(1)}$, $\mathcal{F}_{(2)}$ and $\mathcal{F}_{(3)}$ respectively [7]. The exact equations are of the form

$$\mathcal{F}_{(j)} = V^{(j)} \cdot S_{(j)} \cdot (V^{\mathrm{mod}(j+1,3)} \otimes V^{\mathrm{mod}(j+2,3)})^T \tag{2}$$

where $1 \leq j \leq 3$. This representation in terms on tensor unfolding is equivalent to the original formulation of HOSVD from Equation 1. However, the complexity of the SVD computations for $K \times K$ matrices is $\mathcal{O}(K^3)$. For computational speed, we impose the constraint that $K \leq 8$. The patches from $\mathcal{F}$ are then projected onto the HOSVD transform. The parameter for thresholding the transform coefficients is picked to be $\sigma\sqrt{2 \log p^2 K}$, which is the near-optimal threshold for hard thresholding of the coefficients of a noise-corrupted data vector projected onto any orthonormal basis assuming i.i.d. additive $\mathcal{N}(0, \sigma)$ noise [8]. The complete stack $\mathcal{F}$ is then reconstructed after inverting the transform, thereby filtering all the individual patches. The procedure is repeated over all pixels in sliding window fashion with simple averaging of the multipled hypotheses that appear at any pixel. Note that we filter *all* the individual patches in the ensemble and not just the reference patch. Moreover, we perform denoising using hard thresholding of coefficients as opposed to low-rank matrix approximations because it is difficult to relate the optimal matrix rank to the noise statistics. Some papers such as [10] penalize the matrix nuclear norm for denoising, but this requires iterated optimization for each patch stack with some heuristically chosen parameters.

The aforementioned framework for image denoising is extended to video denoising in the following manner. The search for patches that are similar to a reference patch in a given frame at time instant t_o is performed over all patches in the time frame $[t_o - \Delta, t_o + \Delta]$ where Δ is a temporal search radius. The existence of multiple images of the same scene varying smoothly with respect to one another, yields us greater redundancy which can be exploited for the purpose of better denoising (modulo limitations of computing time).

2.2 Choice of Patch Similarity Measure

Given a reference patch P_n of size $s \times s$ in I_n, we can compute the patches similar to it by using a distance threshold τ_d and selecting all patches P_{ni} such that $\|P_n - P_{ni}\|^2 < \tau_d$. Assuming a fixed, known noise model - $\mathcal{N}(0, \sigma)$, if P_{ni} and P_i were different noisy versions of the same underlying patch P_t (i.e. $P_i \sim \mathcal{N}(P_t, \sigma)$ and $P_{ni} \sim \mathcal{N}(P_t, \sigma)$), the following random variable would have a $\chi^2(s^2)$ distribution:

$$x = \sum_{k=1}^{s^2} \frac{(P_{ref,k} - P_{ik})^2}{2\sigma^2}. \tag{3}$$

The cumulative of a $\chi^2(z)$ random variable is given by

$$F(x; z) = \gamma(\frac{x}{2}, \frac{z}{2}) \tag{4}$$

where $\gamma(x, a)$ stands for the incomplete gamma function defined as

$$\gamma(x, a) = \frac{1}{\Gamma(a)} \int_{t=0}^{x} e^{-t} t^{a-1} dt \tag{5}$$

with $\Gamma(a) = \int_0^\infty e^{-t} t^{(a-1)} dt$ being the Gamma function. We observe that if $z \geq 3$, for any $x \geq 3z$, we have $F(x; z) \geq 0.99$. Therefore for a patch-size of $s \times s$ and under the given σ, we choose $\tau_d = 6\sigma^2 s^2$. Hence we regard two patches to be similar if and only if their mean squared difference was less than or equal to τ_d. Note however, that in our specific implementation, we always restrict the number of similar patches to a maximum of $K = 8$ for the sake of efficiency.

2.3 HOSVD and Universal 3D Transforms

We explain an important theoretical difference between the HOSVD and universal 3D transforms such as a 3D-DCT, 3D-FFT or a product of 2D-DCT and 1D Haar wavelet (as in [6]), in the context of denoising of patch stacks. The latter group of transforms treats the 3D stack as an actual 3D signal - in other words, it assumes a continuity between pixels at corresponding locations in the different patches of the 3D stack. However, as the stack consists of a group of patches similar to the reference patch, all from different locations in the video, this is not a valid assumption. Moreover, a change in the ordering of the patches could potentially affect the denoising results. In the case of HOSVD, permuting the location of the patches in the 3D stack will leave the transform coefficients unchanged (upto a permutation) and hence not affect the denoising results. Summarily, while 3D transforms will enforce (functional) smoothness in the third dimension of the stack, the HOSVD uses statistical criteria for coupled filtering of all the patches from the stack.

3 Experimental Results and Comparisons

We now present experimental results on video denoising. Our dataset consists of the eight well-known gray-scale video sequences available on `http://telin.ugent.be/vzlokoli/PHD/Greyscale/`. Some sequences such as 'Miss America' contain highly homogenous image frames, whereas others such as 'flower' or 'tennis' are quite textured. We tested our denoising algorithm on each sequence for noise from $\mathcal{N}(0, \sigma)$ where $\sigma \in \{20, 25, 30, 35\}$. The quality metric used for evaluation was the PSNR which is computed as $10 \log_{10} \frac{255^2}{\text{MSE}}$ where MSE is the 'mean-squared error'. The results produced by our HOSVD method were compared to those produced by the video version of BM3D [5]. The latter is a two stage algorithm. The first step performs collaborative hard thresholding of the wavelet transform coefficients of a (3D) stack of similar patches. We refer to this step as 'VBM3D-1'. The second step performs collaborative Wiener filtering where the transform domain coefficients are attenuated using ratios computed from patches from the output of 'VBM3D-1'. We refer to this second step as 'VBM3D-2'. The HOSVD algorithm was run with 8×8 patches, a spatial search window of radius 8 and a temporal search radius of 4, for finding similar patches. VBM3D-1 and VBM3D-2 were run using the package provided by the authors of [5] using their default parameter settings (which also

included patch sizes of 8×8). In all experiments, noise was added to the original sequence using the Matlab command noisy = orig + randn(size(orig))*sigma, followed by clipping of the values in the noisy signal to the [0,255] range.

The comparative results between HOSVD, VBM3D-1 and VBM3D-2 are shown in Tables 3 and 3. From these tables, we see that HOSVD produces PSNR values that are superior to VBM3D-1 on most sequences except 'Miss America' which contains highly homogenous frames. The PSNR values produced by HOSVD are usually close behind those of VBM3D-2. On some complex sequences such as 'bus', HOSVD produces results slightly superior to those by VBM3D-2. In general, the difference between the PSNR values is small for the more difficult and textured sequences. We believe that superior results could be produced by HOSVD on homogenous image frames if larger patch sizes were used. *A point to note is that the PSNR values for VBM3D we have reported are slightly less than those reported by the authors of [5] on their webpage* `http://www.cs.tut.fi/foi/GCF-BM3D/`. *We observed that this difference is due to the clipping of the noisy videos to the [0,255] range.*

An interesting point to note is the performance on the 'tennis' sequence which contains large textured regions (on the wall behind the table tennis table) in many frames. Although VBM3D-2 produces a superior PSNR in comparison to HOSVD at all four noise levels, we observed that HOSVD did a much better job than both VBM3D-1 and VBM3D-2 in preserving the texture on the wall. We believe that the fixed bases used by VBM3D-1 (DCT/Haar) have a tendency to wipe out some subtle textures. These textures get further attenuated during the Wiener filtering step in VBM3D-2. These results can be observed in Figure 2 and Table 3. Essentially, this example yet again highlights the advantages of learning the bases *in situ* from noisy data as opposed to using fixed, universal bases.

A similar phenomenon was noted by [11] on the same video sequence (see Figure 4 of [11]). The method in [11] produces a PSNR of 30.21 for $\sigma = 20$, whereas ours produces 30.31. It should be noted that the former makes explicit use of a robust motion estimator as well as a much more sophisticated and global search algorithm for finding similar patches (unlike our method which looks for similar patches in a restricted search radius around the top left corner of the reference patch). This highlights the good denoising properties of the HOSVD bases as well as the benefits of the additional smoothing afforded by patch-based algorithms, unlike [11] which uses a pixel-based algorithm such as (a slightly modified version of) NL-Means [2] for the smoothing. We believe that the exact benefit of motion estimation for denoising videos affected by i.i.d. noise remains a debatable matter (even more so, given the error-prone nature of optical flow computations especially in noisy data, the parameter selection involved and the computational cost), but we leave a rigorous testing of this issue to future work. From the arguments in [11], it does seem that motion estimation prior to denoising will enhance the overall performance in case of *structured* noise which affects real-world color videos.

We show a few more results of the HOSVD method - on frame 75 from the bus and flower sequences at $\sigma = 25$ in Figures 3 and 4 respectively. We have uploaded sample video results (in the form of avi files) on four sequences: coastguard, flower, bus and tennis, each at noise levels 20, 25, 30, 35, on the following webpages: `https://sites.google.com/site/emmcvpr2011submission26/videos` and `https://sites.google.com/site/emmcvprsubmission26part2/emmcvprsubmission26part2`.

Table 1: PSNR results for video sequences for $\sigma \in \{20, 25\}$

Sequence	HOSVD $\sigma = 20$	VBM3D-1 $\sigma = 20$	VBM3D-2 $\sigma = 20$	HOSVD $\sigma = 25$	VBM3D-1 $\sigma = 25$	VBM3D-2 $\sigma = 25$
salesman	32.046	31.73	33.49	30.396	30.298	31.99
bus	30.237	28.69	29.568	28.934	27.524	28.4
flower	28.142	27.436	28.17	26.88	26.038	26.873
miss america	34.32	35.809	37.72	32.737	34.795	36.838
foreman	32.4	32.113	33.37	30.886	30.828	32.148
tennis	30.31	30.04	30.94	29.034	28.733	29.51
coastguard	31.09	30.63	31.75	29.634	29.348	30.54
bicycle	33.09	32.8	34.15	31.453	31.4	32.77

Table 2: PSNR results for 'tennis' sequence for $\sigma = 20$

Sequence	HOSVD	VBM3D-1	VBM3D-2	[11]
tennis	30.31	30.04	30.94	30.21

Table 3: PSNR results for video sequences for $\sigma \in \{30, 35\}$

Sequence	HOSVD $\sigma = 30$	VBM3D-1 $\sigma = 30$	VBM3D-2 $\sigma = 30$	HOSVD $\sigma = 35$	VBM3D-1 $\sigma = 35$	VBM3D-2 $\sigma = 35$
salesman	29.04	29.124	30.66	27.87	28.129	29.486
bus	27.81	26.53	27.45	26.817	25.793	26.66
flower	25.69	24.787	25.727	24.588	23.9	24.81
miss america	31.35	33.91	35.847	30.087	33.05	34.736
foreman	29.645	29.78	31.097	28.594	28.9	30.236
tennis	28	27.8	28.55	27.117	27.076	27.841
coastguard	28.4	28.28	29.497	27.346	27.384	28.628
bicycle	30	30.185	31.476	28.864	29.107	30.28

4 Conclusion

We have presented a very simple algorithm for video denoising and compared it to state of the art methods such as BM3D. Our algorithm yields good results on video denoising despite the fact that it does not employ motion estimation as in [11], or Wiener filtering as in [5]. The algorithm can be easily parallelized for improving efficiency. It sometimes preserves fine textural details better than VBM3D-2 - the current state of the art algorithm in video denoising. The performance of the algorithm could perhaps be improved using: (1) a better and more global search method for finding similar patches, and (2) a robust and efficient method for motion estimation prior to denoising.

References

[1] E. Balster, Y. Zheng, and R. Ewing. Combined spatial and temporal domain wavelet shrinkage algorithm for video denoising. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(2):220–230, Feb. 2006.

[2] A. Buades, B. Coll, and J.M. Morel. Nonlocal image and movie denoising. *Int. J. Comput. Vis.*, 76(2):123–139, 2008.

[3] A. Buades, Y. Lou, J.M. Morel, and Z. Tang. A note on multi-image denoising. In *Local and Non-Local Approximation in Image Processing*, pages 1–15, 2009.

[4] P. Chatterjee and P. Milanfar. Clustering-based denoising with locally learned dictionaries. *IEEE Trans. Image Process.*, 18(7):1438–1451, 2009.

[5] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3D transform-domain collaborative filtering. In *European Signal Processing Conference, EUSIPCO*, 2007.

[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16(8):2080–2095, 2007.

[7] L. de Lathauwer. *Signal Processing Based on Multilinear Algebra*. PhD thesis, Katholieke Universiteit Leuven, Belgium, 1997.

[8] D. Donoho and I. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81:425–455, 1993.

[9] M. Elad and M. Aharon. Image denoising via learned dictionaries and sparse representation. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 17–22, 2006.

[10] H. Ji, C. Liu, Z. Shen, and Y. Xu. Robust video denoising using low rank matrix completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[11] C. Liu and W. Freeman. A high-quality video denoising algorithm based on reliable motion estimation. In *European Conference on Computer Vision*, 2010.

[12] D. Muresan and T. Parks. Adaptive principal components and image denoising. In *IEEE Int. Conf. Image Process.*, pages 101–104, 2003.

[13] A. Rajwade, A. Rangarajan, and A. Banerjee. Image denoising using the higher order singular value decomposition. Technical Report REP-2011-515, Department of CISE, University of Florida, Gainesville, Florida, February 2011.

[14] I. Selesnick and K. Li. Video denoising using 2D and 3D dual-tree complex wavelet transforms. In *SPIE Proceedings of Wavelet Applications in Signal and Image Processing*, 2003.

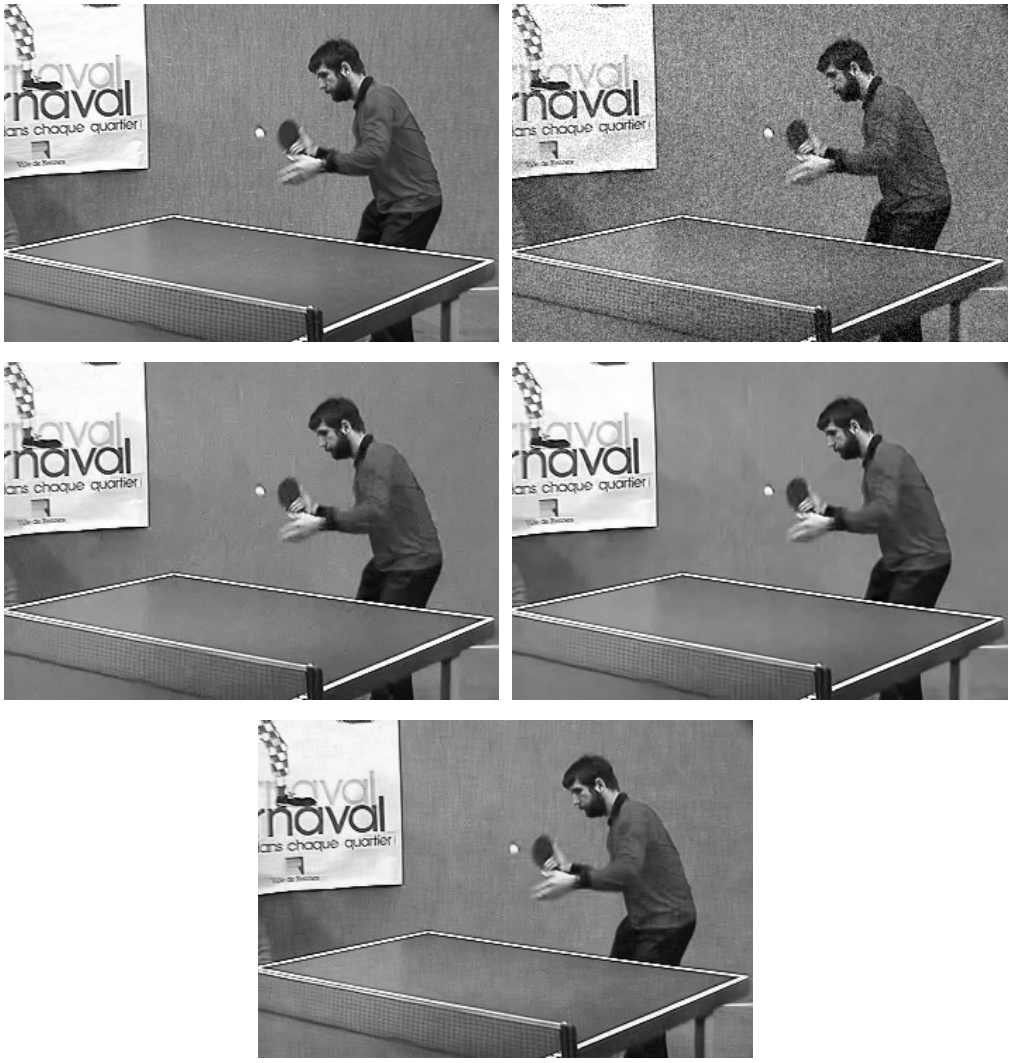

Figure 2: Frame 75 from (a) original 'tennis' sequence, (b) noisy sequence ($\sigma = 20$), (c) sequence denoised by VBM3D-1, (d) sequence denoised by VBM3D-2, (e) sequence denoised by HOSVD. Zoom into the pdf for a better view.

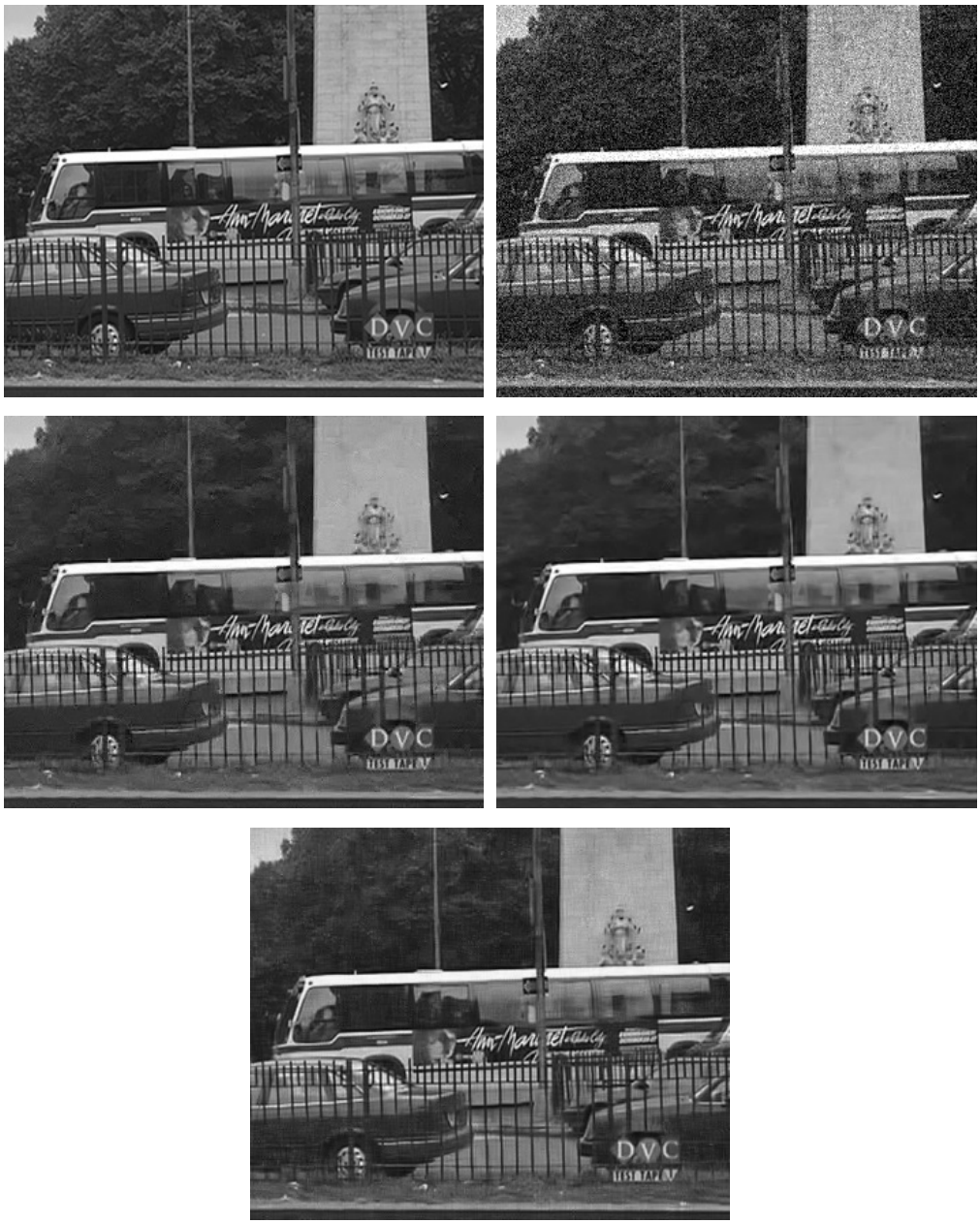

Figure 3: Frame 75 from (a) original 'bus' sequence, (b) noisy sequence ($\sigma = 25$), (c) sequence denoised by VBM3D-1, (d) sequence denoised by VBM3D-2, (e) sequence denoised by HOSVD. Zoom into the pdf for a better view.

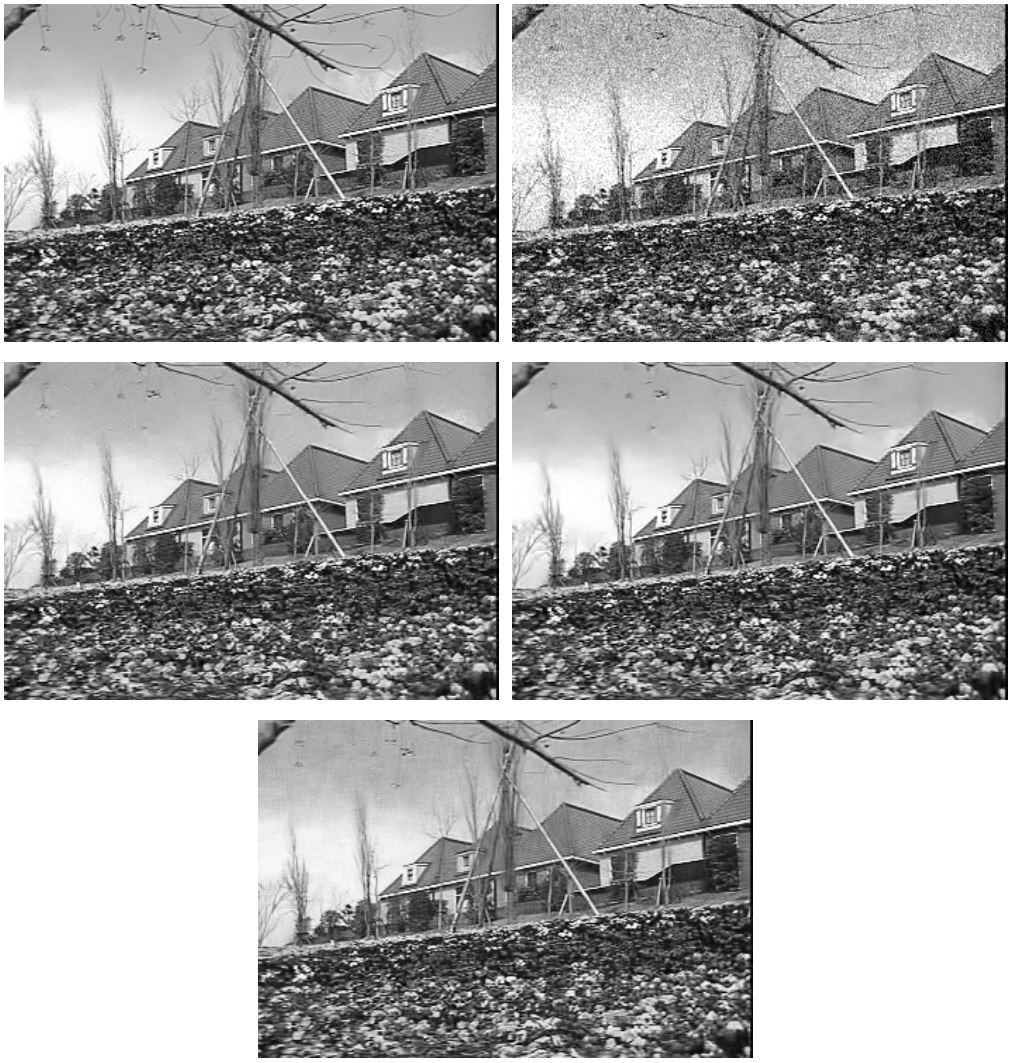

Figure 4: Frame 75 from (a) original 'flower' sequence, (b) noisy sequence ($\sigma = 25$), (c) sequence denoised by VBM3D-1, (d) sequence denoised by VBM3D-2, (e) sequence denoised by HOSVD. Zoom into the pdf for a better view.