

Video Denoising, Deblocking and Enhancement Through Separable 4-D Nonlocal Spatiotemporal Transforms

Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, Karen Egiazarian

Abstract—We propose a powerful video filtering algorithm that exploits temporal and spatial redundancy characterizing natural video sequences. The algorithm implements the paradigm of nonlocal grouping and collaborative filtering, where a higher-dimensional transform-domain representation of the observations is leveraged to enforce sparsity and thus regularize the data: 3-D spatiotemporal volumes are constructed by tracking blocks along trajectories defined by the motion vectors. Mutually similar volumes are then grouped together by stacking them along an additional fourth dimension, thus producing a 4-D structure, termed group, where different types of data correlation exist along the different dimensions: local correlation along the two dimensions of the blocks, temporal correlation along the motion trajectories, and nonlocal spatial correlation (i.e. self-similarity) along the fourth dimension of the group. Collaborative filtering is then realized by transforming each group through a decorrelating 4-D separable transform and then by shrinkage and inverse transformation. In this way, the collaborative filtering provides estimates for each volume stacked in the group, which are then returned and adaptively aggregated to their original positions in the video. The proposed filtering procedure addresses several video processing applications, such as denoising, deblocking, and enhancement of both grayscale and color data. Experimental results prove the effectiveness of our method in terms of both subjective and objective visual quality, and shows that it outperforms the state of the art in video denoising.

Index Terms—Video filtering, video denoising, video deblocking, video enhancement, nonlocal methods, adaptive transforms, motion estimation.

I. INTRODUCTION

SEVERAL factors such as noise, blur, blocking, ringing, and other acquisition or compression artifacts, typically impair digital video sequences. The large number of practical applications involving digital videos has motivated a significant interest in restoration or enhancement solutions, and the literature contains a plethora of such algorithms (see [3], [4] for a comprehensive overview).

At the moment, the most effective approach in restoring images or video sequences exploits the redundancy given by

Matteo Maggioni, Alessandro Foi and Karen Egiazarian are with the Department of Signal Processing, Tampere University of Technology, Finland. Giacomo Boracchi is with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy

This paper is based on and extends the authors' preliminary conference publications [1], [2]

This work was supported by the Academy of Finland (project no. 213462, Finnish Programme for Centres of Excellence in Research 20062011, project no. 252547, Academy Research Fellow 20112016, and project no. 129118, Postdoctoral Researchers Project 20092011), and by Tampere Graduate School in Information Science and Engineering (TISE).

the *nonlocal* similarity between patches at different locations within the data [5], [6]. Algorithms based on this approach have been proposed for various signal-processing problems, and mainly for image denoising [4], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. Specifically, in [7] has been introduced an adaptive pointwise image filtering strategy, called *non-local means*, where the estimate of each pixel x_i is obtained as a weighted average of, in principle, all the pixels x_j of the noisy image, using a family of weights proportional to the similarity between two neighborhoods centered at x_i and x_j . So far, the most effective image-denoising algorithm is BM3D [10], [6], which relies on the so-called grouping and collaborative filtering paradigm: the observation is processed in a blockwise manner and mutually similar 2-D image blocks are stacked into a 3-D group (grouping), which is then filtered through a transform-domain shrinkage (collaborative filtering), simultaneously providing different estimates for each grouped block. These estimates are then returned to their respective locations and eventually aggregated resulting in the denoised image. In doing so, BM3D leverages the spatial correlation of natural images both at the nonlocal and local level, due to the abundance of mutually similar patches and to the high correlation of image data within each patch, respectively. The BM3D filtering scheme has been successfully applied to video denoising in our previous work, V-BM3D [11], as well as to several other applications including image and video super-resolution [14], [15], [16], image sharpening [13], and image deblurring [17].

In V-BM3D, groups are 3-D arrays of mutually similar blocks extracted from a set of consecutive frames of the video sequence. A group may include multiple blocks from the same frame, naturally exploiting in this way the nonlocal similarity characterizing images. However, it is typically along the temporal dimension that most mutually similar blocks can be found. It is well known that motion-compensated videos [18] are extremely smooth along the temporal axis and this fact is exploited by nearly all modern video-coding techniques. Furthermore, experimental analysis in [12] shows that, even when fast motion is present, the similarity along the motion trajectories is much stronger than the nonlocal similarity existing within an individual frame. In spite of this, in V-BM3D the blocks are grouped regardless of whether their similarity comes from the motion tracking over time or the nonlocal spatial content. Consequently, during the filtering, V-BM3D is not able to distinguish between temporal and spatial nonlocal similarity. We recognize this as a conceptual as well

as practical weakness of the algorithm. As a matter of fact, the simple experiments reported in Section VIII demonstrate that the denoising quality do not necessarily increase with the number of spatially self-similar blocks in each group; in contrast, the performances are always improved by exploiting the temporal correlation of the video.

This work proposes V-BM4D, a novel video-filtering approach that, to overcome the above weaknesses, separately exploits the temporal and spatial redundancy of the video sequences. The core element of V-BM4D is the spatiotemporal volume, a 3-D structure formed by a sequence of blocks of the video following a specific trajectory (obtained, for example, by concatenating motion vectors along time) [19], [20]. Thus, contrary to V-BM3D, V-BM4D does not group blocks, but mutually similar spatiotemporal volumes according to a nonlocal search procedure. Hence, groups in V-BM4D are 4-D stacks of 3-D volumes, and the collaborative filtering is then performed via a separable 4-D spatiotemporal transform. The transform leverages the following three types of correlation that characterize natural video sequences: local spatial correlation between pixels in each block of a volume, local temporal correlation between blocks of each volume, and nonlocal spatial and temporal correlation between volumes of the same group. The 4-D group spectrum is thus highly sparse, which makes the shrinkage more effective than in V-BM3D, yielding superior performance of V-BM4D in terms of noise reduction.

In this work we extend the basic implementation of V-BM4D as a grayscale denoising filter introduced in the conference paper [1] presenting its modifications for the deblocking and deringing of compressed videos, as well as for the enhancement (sharpening) of low-contrast videos. Then, leveraging the approach presented in [10], [21], we generalize V-BM4D to perform collaborative filtering of color (multi-channel) data. An additional, and fundamental, contribution of this paper is an experimental analysis of the different types of correlation characterizing video data, and how these affect the filtering quality.

The paper is organized as follows. Section II introduces the observation model, the formal definitions, and describes the fundamental steps of V-BM4D, while Section III discusses the implementation aspects, with particular emphasis on the computation of motion vectors. The application of V-BM4D to deblocking and deringing is given in Section IV, where it is shown how to compute the thresholds used in the filtering from the compression parameters of a video; video enhancement (sharpening) is presented in Section V. Before the conclusions, we provide a comprehensive collection of experiments and a discussion of the V-BM4D performance in Section VI, and a detailed analysis of its computational complexity in Section VII.

II. BASIC ALGORITHM

The aim of the proposed algorithm is to provide an estimate of the original video from the observed data. For the algorithm design, we assume the common additive white Gaussian noise model.

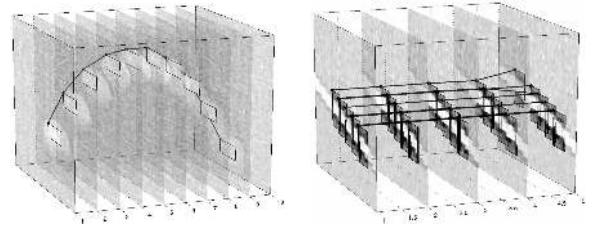


Fig. 1. Illustration of a trajectory and the associated volume (left), and a group of mutually similar volumes (right). These have been calculated from the sequence *Tennis* corrupted by white Gaussian noise with $\sigma = 20$.

A. Observation Model

We consider the observed video as a noisy image sequence $z : X \times T \rightarrow \mathbb{R}$ defined as

$$z(\mathbf{x}, t) = y(\mathbf{x}, t) + \eta(\mathbf{x}, t), \quad \mathbf{x} \in X, t \in T, \quad (1)$$

where y is the original (unknown) video, $\eta(\cdot, \cdot) \sim \mathcal{N}(0, \sigma^2)$ is i.i.d. white Gaussian noise, and (\mathbf{x}, t) are the 3-D spatiotemporal coordinates belonging to the spatial domain $X \subset \mathbb{Z}^2$ and time domain $T \subset \mathbb{Z}$, respectively. The frame of the video z at time t is denoted by $z(X, t)$.

The V-BM4D algorithm comprises three fundamental steps inherited from the BM3D paradigm, specifically grouping (Section II-C), collaborative filtering (Section II-D) and aggregation (Section II-E). These steps are performed for each spatiotemporal volume of the video (Section II-B).

B. Spatiotemporal Volumes

Let $B_z(\mathbf{x}_0, t_0)$ denote a square block of fixed size $N \times N$ extracted from the noisy video z ; without loss of generality, the coordinates (\mathbf{x}_0, t_0) identify the top-left pixel of the block in the frame $z(X, t_0)$. A spatiotemporal volume is a 3-D sequence of blocks built following a specific trajectory along time, which is supposed to follow the motion in the scene. Formally, the trajectory associated to (\mathbf{x}_0, t_0) is defined as

$$\text{Traj}(\mathbf{x}_0, t_0) = \left\{ (\mathbf{x}_j, t_0 + j) \right\}_{j=-h^-}^{h^+}, \quad (2)$$

where the elements $(\mathbf{x}_j, t_0 + j)$ are time-consecutive coordinates, each of these represents the position of the reference block $B_z(\mathbf{x}_0, t_0)$ within the neighboring frames $z(X, t_0 + j)$, $j = -h^-, \dots, h^+$. For the sake of simplicity, in this section it is assumed $h^- = h^+ = h$ for all $(\mathbf{x}, t) \in X \times T$.

The trajectories can be either directly computed from the noisy video, or, when a coded video is given, they can be obtained by concatenating motion vectors. In what follows we assume that, for each $(\mathbf{x}_0, t_0) \in X \times T$, a trajectory $\text{Traj}(\mathbf{x}_0, t_0)$ is given and thus the 3-D spatiotemporal volume associated to (\mathbf{x}_0, t_0) can be determined as

$$V_z(\mathbf{x}_0, t_0) = \{ B_z(\mathbf{x}_i, t_i) : (\mathbf{x}_i, t_i) \in \text{Traj}(\mathbf{x}_0, t_0) \}, \quad (3)$$

where the subscript z specifies that the volumes are extracted from the noisy video.

C. Grouping

Groups are stacks of mutually similar volumes and constitute the nonlocal element of V-BM4D. Mutually similar volumes are determined by a nonlocal search procedure as in [10].

Specifically, let $\text{Ind}(\mathbf{x}_0, t_0)$ be the set of indices identifying those volumes that, according to a distance operator δ^v , are similar to $V_z(\mathbf{x}_0, t_0)$:

$$\text{Ind}(\mathbf{x}_0, t_0) = \{(\mathbf{x}_i, t_i) : \delta^v(V_z(\mathbf{x}_0, t_0), V_z(\mathbf{x}_i, t_i)) < \tau_{\text{match}}\}.$$

The parameter $\tau_{\text{match}} > 0$ controls the minimum degree of similarity among volumes with respect to the distance δ^v , which is typically the ℓ^2 -norm of the difference between two volumes.

The group associated to the reference volume $V_z(\mathbf{x}_0, t_0)$ is then

$$G_z(\mathbf{x}_0, t_0) = \{V_z(\mathbf{x}_i, t_i) : (\mathbf{x}_i, t_i) \in \text{Ind}(\mathbf{x}_0, t_0)\}. \quad (4)$$

In (4) we implicitly assume that the 3-D volumes are stacked along a fourth dimension; hence the groups are 4-D data structures. The order of the spatiotemporal volumes in the 4-D stacks is based on their similarity with the reference volume. Note that since $\delta^v(V_z, V_z) = 0$, every group $G_z(\mathbf{x}_0, t_0)$ contains, at least, its reference volume $V_z(\mathbf{x}_0, t_0)$. Figure 1 shows an example of trajectories and volumes belonging to a group.

D. Collaborative Filtering

According to the general formulation of the grouping and collaborative-filtering approach for a d -dimensional signal [10], groups are $(d+1)$ -dimensional structures of similar d -dimensional elements, which are then jointly filtered. In particular, each of the grouped elements influences the filtered output of all the other elements of the group: this is the basic idea of collaborative filtering. It is typically realized through the following steps: firstly a $(d+1)$ -dimensional separable linear transform is applied to the group, then the transformed coefficients are shrunk, for example by hard thresholding or by Wiener filtering, and finally the $(d+1)$ -dimensional transform is inverted to obtain an estimate for each grouped element.

The core elements of V-BM4D are the spatiotemporal volumes ($d=3$), and thus the collaborative filtering performs a 4-D separable linear transform \mathcal{T}_{4D} on each 4-D group $G_z(\mathbf{x}_0, t_0)$, and provides an estimate for each grouped volume V_z :

$$\hat{G}_y(\mathbf{x}_0, t_0) = \mathcal{T}_{4D}^{-1}(\Upsilon(\mathcal{T}_{4D}(G_z(\mathbf{x}_0, t_0)))) ,$$

where Υ denotes a generic shrinkage operator. The filtered 4-D group $\hat{G}_y(\mathbf{x}_0, t_0)$ is composed of volumes $\hat{V}_y(\mathbf{x}, t)$

$$\hat{G}_y(\mathbf{x}_0, t_0) = \{\hat{V}_y(\mathbf{x}_i, t_i) : (\mathbf{x}_i, t_i) \in \text{Ind}(\mathbf{x}_0, t_0)\},$$

with each \hat{V}_y being an estimate of the corresponding unknown volume V_y in the original video y .

E. Aggregation

The groups \hat{G}_y constitute a very redundant representation of the video, because in general the volumes \hat{V}_y overlap and, within the overlapping parts, the collaborative filtering provides multiple estimates at the same coordinates (\mathbf{x}, t) . For this reason, the estimates are aggregated through a convex combination with adaptive weights. In particular, the estimate \hat{y} of the original video is computed as

$$\hat{y} = \frac{\sum_{(\mathbf{x}_0, t_0) \in X \times T} \left(\sum_{(\mathbf{x}_i, t_i) \in \text{Ind}(\mathbf{x}_0, t_0)} w(\mathbf{x}_0, t_0) \hat{V}_y(\mathbf{x}_i, t_i) \right)}{\sum_{(\mathbf{x}_0, t_0) \in X \times T} \left(\sum_{(\mathbf{x}_i, t_i) \in \text{Ind}(\mathbf{x}_0, t_0)} w(\mathbf{x}_0, t_0) \chi(\mathbf{x}_i, t_i) \right)}, \quad (5)$$

where we assume $\hat{V}_y(\mathbf{x}_i, t_i)$ to be zero-padded outside its domain, $\chi(\mathbf{x}_i, t_i) : X \times T \rightarrow \{0, 1\}$ is the characteristic function (indicator) of the support of the volume $\hat{V}_y(\mathbf{x}_i, t_i)$, and the aggregation weights $w(\mathbf{x}_0, t_0)$ are different for different groups. Aggregation weights may depend on the result of the shrinkage in the collaborative filtering, and these are typically defined to be inversely proportional to the total sample variance of the estimate of the corresponding groups [10]. Intuitively, the sparser is the shrunk 4-D spectrum $\hat{G}_y(\mathbf{x}_0, t_0)$, the larger is the corresponding weight $w(\mathbf{x}_0, t_0)$. Such aggregation is a well-established procedure to obtain a global estimate from different overlapping local estimates [22], [23].

III. IMPLEMENTATION ASPECTS

A. Computation of the Trajectories

In our implementation of V-BM4D, we construct trajectories by concatenating motion vectors which are defined as follows.

1) *Location prediction*: As far as two consecutive spatiotemporal locations $(\mathbf{x}_{i-1}, t_i - 1)$ and (\mathbf{x}_i, t_i) of a block are known, we can define the corresponding motion vector (velocity) as $\mathbf{v}(\mathbf{x}_i, t_i) = \mathbf{x}_{i-1} - \mathbf{x}_i$. Hence, under the assumption of smooth motion, we can predict the position $\hat{\mathbf{x}}_i(t_i + 1)$ of the block in the frame $z(X, t_i + 1)$ as

$$\hat{\mathbf{x}}_i(t_i + 1) = \mathbf{x}_i + \gamma_p \cdot \mathbf{v}(\mathbf{x}_i, t_i), \quad (6)$$

where $\gamma_p \in [0, 1]$ is a weighting factor of the prediction. In the case $(\mathbf{x}_{i-1}, t_i - 1)$ is not available, we consider the lack of motion as the most likely situation and we set $\hat{\mathbf{x}}_i(t_i + 1) = \mathbf{x}_i$. Analogous predictions can be made when looking for precedent blocks in the sequence.

2) *Similarity criterion*: The motion of a block is generally tracked by identifying the most similar block in the subsequent or precedent frame. However, since we deal with noisy signals, it is advisable to enforce motion-smoothness priors to improve the tracking. In particular, given the predicted future $\hat{\mathbf{x}}_i(t_i + 1)$ or past $\hat{\mathbf{x}}_i(t_i - 1)$ positions of the block $B_z(\mathbf{x}_i, t_i)$, we define the similarity between $B_z(\mathbf{x}_i, t_i)$ and $B_z(\mathbf{x}_j, t_i \pm 1)$, through a penalized quadratic difference

$$\delta^b(B_z(\mathbf{x}_i, t_i), B_z(\mathbf{x}_j, t_i \pm 1)) = \frac{\|B_z(\mathbf{x}_i, t_i) - B_z(\mathbf{x}_j, t_i \pm 1)\|_2^2}{N^2} + \gamma_d \|\hat{\mathbf{x}}_i(t_i \pm 1) - \mathbf{x}_j\|_2, \quad (7)$$

where $\hat{\mathbf{x}}_i(t_i \pm 1)$ is defined as in (6), and $\gamma_d \in \mathbb{R}^+$ is the penalization parameter. Observe that the tracking is performed separately in time $t_i + 1$ and $t_i - 1$.

V-BM4D constructs the trajectory (2) by repeatedly minimizing (7). Formally, the motion of $B_z(\mathbf{x}_i, t_i)$ from time t_i to $t_i \pm 1$ is determined by the position $\mathbf{x}_{i \pm 1}$ that minimizes (7) as

$$\mathbf{x}_{i \pm 1} = \arg \min_{\mathbf{x}_k \in \mathcal{N}_i} \left\{ \delta^b(B_z(\mathbf{x}_i, t_i), B_z(\mathbf{x}_k, t_i \pm 1)) \right\},$$

where \mathcal{N}_i is an adaptive spatial search neighborhood in the frame $z(X, t_i \pm 1)$ (further details are given in Section III-A3). Even though such $\mathbf{x}_{i \pm 1}$ can be always found, we stop the trajectory construction whenever the corresponding minimum distance δ^b exceeds a fixed parameter $\tau_{\text{traj}} \in \mathbb{R}^+$, which imposes a minimum amount of similarity along the spatiotemporal volumes. This allows V-BM4D to effectively

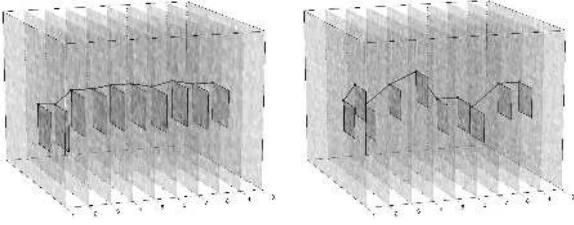


Fig. 2. Effect of different penalties $\gamma_d = 0.025$ (left) and $\gamma_d = 0$ (right) on the background textures of the sequence *Tennis* corrupted by Gaussian noise with $\sigma = 20$. The block positions at time $t = 1$ are the same in both experiments.

deal with those situations, such as occlusions and changes of scene, where consistent blocks (in terms of both similarity and motion smoothness) cannot be found.

Figure 2 illustrates two trajectories estimated using different penalization parameters γ_d . Observe that the penalization term becomes essential when blocks are tracked within flat areas or homogeneous textures in the scene. In fact, the right image of Figure 2 shows that without a position-dependent distance metric the trajectories would be mainly determined by the noise. As a consequence, the collaborative filtering would be less effective because of the badly conditioned temporal correlation of the data within the volumes.

3) *Search neighborhood*: Because of the penalty term $\gamma_d \|\hat{\mathbf{x}}_i(t_i \pm 1) - \mathbf{x}_j\|_2$, the minimizer of (7) is likely close to $\hat{\mathbf{x}}_i(t_i \pm 1)$. Thus, we can rightly restrict the minimization of (7) to a spatial search neighborhood \mathcal{N}_i centered at $\hat{\mathbf{x}}_i(t_i \pm 1)$. We experienced that it is convenient to make the search-neighborhood size, $N_{PR} \times N_{PR}$, adaptive on the velocity of the tracked block (magnitude of motion vector) by setting

$$N_{PR} = N_S \cdot \left(1 - \gamma_w \cdot e^{-\frac{\|\mathbf{v}(\mathbf{x}_i, t_i)\|_2^2}{2 \cdot \sigma_w^2}} \right),$$

where N_S is the maximum size of \mathcal{N}_i , $\gamma_w \in [0, 1]$ is a scaling factor and $\sigma_w > 0$ is a tuning parameter. As the velocity \mathbf{v} increases, N_{PR} approaches N_S accordingly to σ_w ; conversely, when the velocity is zero $N_{PR} = N_S(1 - \gamma_w)$. By setting a proper value of σ_w we can control the decay rate of the exponential term as a function of \mathbf{v} or, in other words, how permissive is the window contraction with respect to the velocity of the tracked block.

B. Sub-volume Extraction

So far, the number of frames spanned by all the trajectories has been assumed fixed and equal to h . However, because of occlusions, scene changes or heavy noise, any trajectory $\text{Traj}(\mathbf{x}_i, t_i)$ can be interrupted at any time, i.e. whenever the distance between consecutive blocks falls below the threshold τ_{traj} . Thus, given a temporal extent $[t_i - h_i^-, t_i + h_i^+]$ for the trajectory $\text{Traj}(\mathbf{x}_i, t_i)$, we have that in general $0 \leq h_i^- \leq h$ and $0 \leq h_i^+ \leq h$, where h denotes the maximum forward and backward extent of the trajectories (thus of volumes) allowed in the algorithm.

As a result, in principle, V-BM4D may stack together volumes having different lengths. However, in practice, because of the separability of the transform \mathcal{T}_{4D} , every group $G_z(\mathbf{x}_i, t_i)$ has to be composed of volumes having the same

length. Thus, for each reference volume $V_z(\mathbf{x}_0, t_0)$, we only consider the volumes $V_z(\mathbf{x}_i, t_i)$ such that $t_i = t_0$, $h_i^- \geq h_0^-$ and $h_i^+ \geq h_0^+$. Then, we extract from each $V_z(\mathbf{x}_i, t_i)$ the sub-volume having temporal extent $[t_0 - h_0^-, t_0 + h_0^+]$, denoted as $\mathcal{E}_{L_0}(V_z(\mathbf{x}_i, t_i))$. Among all the possible criteria for extracting a sub-volume of length $L_0 = h_0^- + h_0^+ + 1$ from a longer volume, our choice aims at limiting the complexity while maintaining a high correlation within the grouped volumes, because we can reasonably assume that similar objects at different positions are represented by similar volumes along time.

In the grouping, we set as distance operator δ^v the ℓ^2 -norm of the difference between time-synchronous volumes normalized with respect to their lengths:

$$\delta^v(V_z(\mathbf{x}_0, t_0), V_z(\mathbf{x}_i, t_i)) = \frac{\|V_z(\mathbf{x}_0, t_0) - \mathcal{E}_{L_0}(V_z(\mathbf{x}_i, t_i))\|_2^2}{L_0}. \quad (8)$$

C. Two-Stage Implementation with Collaborative Wiener Filtering

The general procedure described in Section II is implemented in two cascading stages, each composed of the grouping, collaborative filtering and aggregation steps.

1) *Hard-thresholding stage*: In the first stage, volumes are extracted from the noisy video z , and groups are then formed using the δ^v -operator (8) and the predefined threshold $\tau_{\text{match}}^{\text{ht}}$. Collaborative filtering is realized by hard thresholding each group $G_z(\mathbf{x}, t)$ in 4-D transform domain:

$$\hat{G}_y^{\text{ht}}(\mathbf{x}, t) = \mathcal{T}_{4D}^{\text{ht}^{-1}} \left(\Upsilon^{\text{ht}} \left(\mathcal{T}_{4D}^{\text{ht}} (G_z(\mathbf{x}_0, t_0)) \right) \right), \quad (\mathbf{x}, t) \in X \times T,$$

where $\mathcal{T}_{4D}^{\text{ht}}$ is the 4-D transform and Υ^{ht} is the hard-threshold operator with threshold $\sigma \lambda_{4D}$.

The outcome of the hard-thresholding stage, \hat{y}^{ht} , is obtained by aggregating with a convex combination all the estimated groups $\hat{G}_y^{\text{ht}}(\mathbf{x}, t)$, as defined in (5). The adaptive weights used in this combination are inversely proportional to the number $N_{(\mathbf{x}_0, t_0)}^{\text{ht}}$ of non-zero coefficients of the corresponding hard-thresholded group $\hat{G}_y^{\text{ht}}(\mathbf{x}_0, t_0)$: that is $w_{(\mathbf{x}_0, t_0)}^{\text{ht}} = 1/N_{(\mathbf{x}_0, t_0)}^{\text{ht}}$, which provides an estimate of the total variance of $\hat{G}_y^{\text{ht}}(\mathbf{x}, t)$. In such a way, we assign larger weights to the volumes belonging to groups having sparser representation in \mathcal{T}_{4D} domain.

2) *Wiener-filtering stage*: In the second stage, the motion estimation is improved by extracting new trajectories $\text{Traj}_{\hat{y}^{\text{ht}}}$ from the basic estimate \hat{y}^{ht} , and the grouping is performed on the new volumes $V_{\hat{y}^{\text{ht}}}$. Volume matching is still performed through the δ^v -distance, but using a different threshold $\tau_{\text{match}}^{\text{wie}}$. The indices identifying similar volumes $\text{Ind}_{\hat{y}^{\text{ht}}}(\mathbf{x}, t)$ are used to construct both groups G_z and $G_{\hat{y}^{\text{ht}}}$, composed by volumes extracted from the noisy video z and from the estimate y^{ht} , respectively.

Collaborative filtering is hence performed using an empirical Wiener filter in $\mathcal{T}_{4D}^{\text{wie}}$ transform domain. Shrinkage is realized by scaling the 4-D transform coefficients of each group $G_z(\mathbf{x}_0, t_0)$, extracted from the noisy video z , with the Wiener attenuation coefficients $\mathbf{W}(\mathbf{x}_0, t_0)$,

$$\mathbf{W}(\mathbf{x}_0, t_0) = \frac{|\mathcal{T}_{4D}^{\text{wie}}(G_{\hat{y}^{\text{ht}}}(\mathbf{x}_0, t_0))|^2}{|\mathcal{T}_{4D}^{\text{wie}}(G_{\hat{y}^{\text{ht}}}(\mathbf{x}_0, t_0))|^2 + \sigma^2},$$

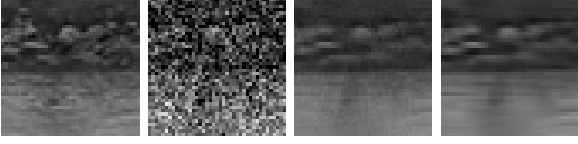


Fig. 3. V-BM4D two stage denoising of the sequence *Coastguard*. From left to right: original video y , noisy video z ($\sigma = 40$), result of the first stage y^{ht} (frame PSNR 28.58 dB) and final estimate y^{w} (frame PSNR 29.38 dB).

that are computed from the energy of the 4-D spectrum of the group $G_{\hat{y}^{\text{ht}}}(\mathbf{x}_0, t_0)$. Eventually, the group estimate is obtained by inverting the 4-D transform as

$$\hat{G}_y^{\text{w}}(\mathbf{x}_0, t_0) = \mathcal{T}_{4D}^{\text{w}^{-1}}(\mathbf{W}(\mathbf{x}_0, t_0) \cdot \mathcal{T}_{4D}^{\text{w}}(G_z(\mathbf{x}_0, t_0))),$$

where \cdot denotes the element-wise product. The final global estimate \hat{y}^{w} is computed by the aggregation (5), using the weights $w_{(\mathbf{x}_0, t_0)}^{\text{w}} = \|\mathbf{W}(\mathbf{x}_0, t_0)\|_2^{-2}$, which follow from considerations similar to those underlying the adaptive weights used in the first stage.

D. Settings

The parameters involved in the motion estimation and in the grouping, that is γ_d , τ_{traj} and τ_{match} , depend on the noise standard deviation σ . Intuitively, in order to compensate the effects of the noise, the larger is σ , the larger become the thresholds controlling blocks and volumes matching. For the sake of simplicity we model such dependencies as second-order polynomials in σ : $\gamma_d(\sigma)$, $\tau_{\text{traj}}(\sigma)$ and $\tau_{\text{match}}(\sigma)$. The nine coefficients required to describe the three polynomials are jointly optimized using the Nelder-Mead simplex direct search algorithm [24], [25]. As optimization criterion, we maximize the sum of the restoration performance (PSNR) of V-BM4D applied over a collection of test videos corrupted by synthetic noise having different values of σ . Namely, we considered *Salesman*, *Tennis*, *Flower Garden*, *Miss America*, *Coastguard*, *Foreman*, *Bus*, and *Bicycle* corrupted by white Gaussian noise having σ levels ranging from 5 and 70. The resulting polynomials are

$$\gamma_d(\sigma) = 0.0005 \cdot \sigma^2 - 0.0059 \cdot \sigma + 0.0400, \quad (9)$$

$$\tau_{\text{traj}}(\sigma) = 0.0047 \cdot \sigma^2 + 0.0676 \cdot \sigma + 0.4564, \quad (10)$$

$$\tau_{\text{match}}(\sigma) = 0.0171 \cdot \sigma^2 + 0.4520 \cdot \sigma + 47.9294. \quad (11)$$

The solid lines in Figure 4 show the above functions. We also plot, using different markers, the best values of the three parameters obtained by unconstrained and independent optimizations of V-BM4D for each test video and value of σ . Empirically, the polynomials demonstrate a good approximation of the optimum (γ_d , τ_{traj} , τ_{match}). Within the considered σ range, the curve (9) is “practically” monotone increasing despite its negative first-degree coefficient. We refrain from introducing additional constraints to the polynomials as well as from considering additional σ values smaller than 5, because the resulting sequences would be mostly affected by the noise and quantization artifacts intrinsic in the original test-data.

During the second stage (namely, the Wiener filtering) the γ_d , τ_{traj} and τ_{match} parameters can be considered as constants and independent, because in the processed sequence \hat{y}^{ht} the noise has been considerably reduced with respect to the observation z ; this is evident when looking at the second and

third image of Figure 3. Moreover, since in this stage both the trajectories and groups are determined from the basic estimate \hat{y}^{ht} , there is no a straightforward relation with σ , the noise standard deviation in z .

IV. DEBLOCKING

Most video compression techniques, such as MPEG-4 [26] or H.264 [27], make use of block-transform coding and thus may suffer, especially at low bitrates, from several compression artifacts such as blocking, ringing, mosquito noise, and flickering. These artifacts are mainly due to the coarse quantization of the block-transform coefficients and to the motion compensation. Moreover, since each block is processed separately, the correlation between pixels at the borders of neighboring blocks is typically lost during the compression, resulting in false discontinuities in the decoded video (such as those shown in the blocky frames in Figure 8).

A large number of deblocking filters have been proposed in the last decade; among them we mention frame-based enhancement using a linear low-pass filter in spatial or transform domain [28], projection onto convex sets (POCS) methods [29], spatial block boundary filter [30], statistical modeling methods [31] or shifted thresholding [32]. Additionally, most of modern video coding block-based techniques, such as H.264 or MPEG-4, embed an in-loop deblocking filter as an additional processing step in the decoder [26].

Inspired by [33], we treat the blocking artifacts as additive noise. This choice allows us to model the compressed video z as in (1), where y now corresponds to the original uncompressed video, and η represents the compression artifacts. In what follows, we focus our attention on MPEG-4 compressed videos. In this way, the proposed filter can be applied reliably over different types of data degradations with little need of adjustment or user intervention.

In order to use V-BM4D as a deblocking filter, we need to determine a suitable value of σ to handle the artifacts in a compressed video. To this purpose, we proceed as in the previous section and we identify the optimum value of σ for a set of test sequences compressed at various rates. Figure 5 shows these optimum values plotted against the average bit-per-pixel (bpp) rate of the compressed video and the parameter q that controls the quantization of the block-transform coefficients [26] (Figure 5(a)). Let us observe that both the bpp and q parameters are easily accessible from any given MPEG-4 coded video. These plots suggest that a power law may conveniently explain the relation between the optimum value of σ and both the bpp rate and q . Hence, we fit such bivariate function to the optimum values via least-squares regression, obtaining the adaptive value of σ for the V-BM4D deblocking filter as

$$\sigma(\text{bpp}, q) = 0.09 \cdot q^{1.11} \cdot \text{bpp}^{-0.46} + 3.37 \quad (12)$$

The function $\sigma(\text{bpp}, q)$ is shown in Figure 5 (right). Note that in MPEG-4 the parameter q ranges from 2 to 31, where higher values correspond to a coarser quantization and consequently lower bitrates. As a matter of fact, when q increases and/or bpp decreases, the optimum σ increases, in order to effectively cope with stronger blocking artifacts. Clearly, a much larger

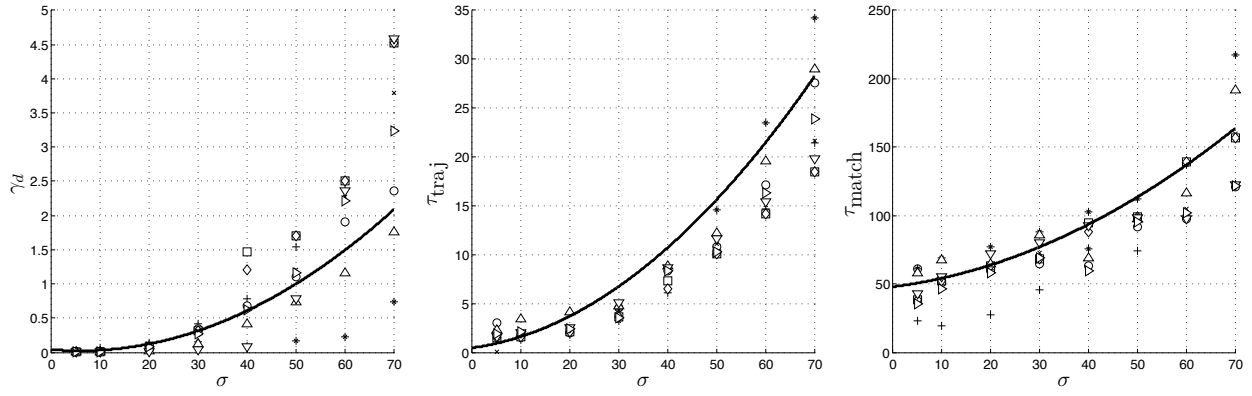


Fig. 4. From left to right, the second-order polynomials (9), (10), and (11) describing the relation between the parameters γ_d , τ_{traj} and τ_{match} and the noise standard deviation σ . The nine coefficients of the three polynomials have been determined by maximizing the sum of the PSNR of the test sequences *Salesman* (+), *Tennis* (o), *Flower Garden* (*), *Miss America* (x), *Coastguard* (□), *Foreman* (◇), *Bus* (Δ), and *Bicycle* (▽), corrupted by white Gaussian noise having σ ranging between 5 and 70. As comparison, we superimpose the optimum parameter for each test sequence and σ .

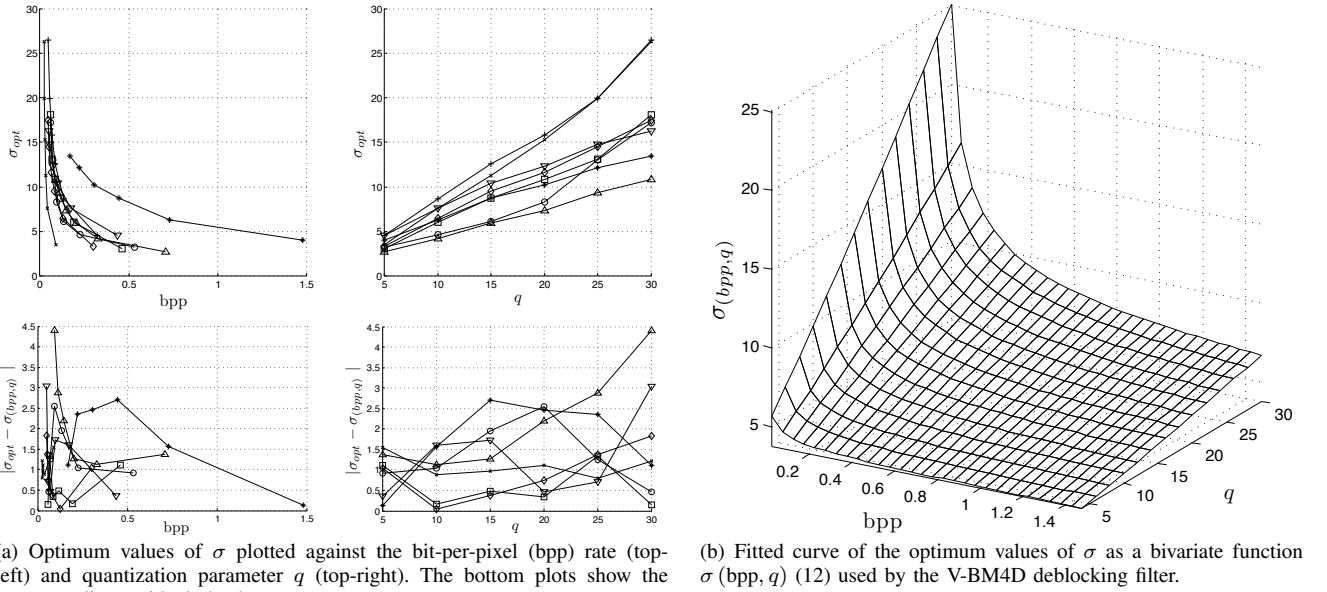


Fig. 5. The sequences used in the fitting are *Salesman* (+), *Tennis* (o), *Flower Garden* (*), *Miss America* (x), *Coastguard* (□), *Foreman* (◇), *Bus* (Δ), and *Bicycle* (▽).

value of σ could result in oversmoothing, while much smaller values may not suffice for effectively reducing the compression artifacts. While in this paper we mostly deal with short test sequences, and we compute the bpp as the average rate over the whole sequence, we argue that in practice this rate should be computed as the average over a limited set of frames, namely the so-called group of pictures (GOP) built around each intra-coded frame. In principle, one could learn a model for σ together with all the remaining V-BM4D parameters at once (possibly achieving better results); but this would have increased the risk of overfitting the many parameters to the peculiarities of this compression method, and would have complicated the optimization task.

Let us remark that V-BM4D deblocking can be straightforwardly applied also to videos compressed by other encoders than MPEG-4, because the q parameter can be both estimated as a subjective quality metric for compressed videos, or as an

objective measurement [34] on the impairing artifacts to be filtered out.

V. ENHANCEMENT

Enhancement is used to improve the video quality, so that the filtered video becomes more pleasing to human subjective judgment and/or better suited for subsequent automatic interpretation tasks, as segmentation or pattern recognition. In particular, by enhancement we refer to the sharpening of degraded details in images (frames) characterized by low contrast.

Among the existing enhancement techniques we mention methods based on histogram manipulation [35], linear and non-linear unsharp masking [36], [37], [38], fuzzy logic [39], and weighted median filter [40], [41]. Transform-domain methods generally apply a nonlinear operator to the transform coefficients of the processed image/video in order to accentuate specific portions of the spectrum, which eventually results

in sharpening of details [42], [43], [35], [13]. One of the most popular technique is alpha-rooting [42], which raises the magnitude of each transform coefficient ϕ_i of the processed spectrum Φ to a power $\frac{1}{\alpha}$, with $\alpha > 1$ as

$$\bar{\phi}_i = \begin{cases} \text{sign}[\phi_i] |\phi_i| \left| \frac{\phi_i}{\phi_0} \right|^{\frac{1}{\alpha}}, & \text{if } \phi_0 \neq 0, \\ \phi_i, & \text{otherwise,} \end{cases} \quad (13)$$

where ϕ_0 is the DC term and $\bar{\phi}_i$ is the resulting sharpened coefficients. Observe that $\alpha > 1$ induces sharpening, as it scales the large coefficients relatively to the small ones, i.e. those carrying high-frequency information [42]. Although (13) assumes real-valued transform coefficients, it can be generalized to complex-valued ones, observing that alpha-rooting preserves the sign in the former case, and the phase in the latter.

A critical issue in enhancement is the amplification of the noise together with the sharpening of image details [44], [42], an effect that becomes more severe as the amount of applied sharpening increases. In order to cope with this problem, a joint application of a denoising and sharpening filter is often recommendable, and in particular this practice has been investigated in [13], [39].

Enhancement of digital videos, following the approach proposed in [13], can be easily performed by combining the V-BM4D filter with the alpha-rooting operator (13), in order to simultaneously reduce the noise and sharpen the original signal. The V-BM4D sharpening algorithm still comprises the grouping, collaborative filtering and aggregation steps, and it is carried out through the hard-thresholding stage only. The alpha-rooting operator is applied on the thresholded coefficients within the collaborative filtering step, before inverting the 4-D transform. Note that, since the alpha-rooting amplifies the group coefficients, the total variance of the filtered group changes, thus the aggregation weights cannot be estimated from the number of retained non-zero coefficients $N_{(\mathbf{x}_0, t_0)}^{\text{har}}$. A simple estimator is devised in [13], and can be used to define the weights of (5) as

$$w_{(\mathbf{x}_0, t_0)}^{\text{har}} = \frac{1}{\sum_{\Phi(i) \neq 0} w_i \sigma^2},$$

having

$$w_i = \left(1 - \frac{1}{\alpha}\right)^2 |\phi_0|^{-\frac{2}{\alpha}} |\phi_i|^{\frac{2}{\alpha}} + \frac{1}{\alpha^2} |\phi_i|^{\frac{2}{\alpha}-2} |\phi_0|^{2-\frac{2}{\alpha}},$$

where Φ is the transformed spectrum of the group $G_z^{\text{ht}}(\mathbf{x}_0, t_0)$ resulting from hard thresholding, and ϕ_0 is its corresponding DC coefficient. The DC-term is not alpha-rooted, thus its contribution to the total variance of the sharpened group should be σ^2 . However, in order to avoid completely flat blocks being awarded with excessively large weights, the weight for the DC-term is set equal to the weight of the smallest retained coefficients, i.e. those having magnitude $\sigma \lambda_{4D}$ as

$$w_0 = \left(1 - \frac{1}{\alpha}\right)^2 |\phi_0|^{-\frac{2}{\alpha}} |\sigma \lambda_{4D}|^{\frac{2}{\alpha}} + \frac{1}{\alpha^2} |\sigma \lambda_{4D}|^{\frac{2}{\alpha}-2} |\phi_0|^{2-\frac{2}{\alpha}}.$$

The separability of the 4-D transform can be exploited to extend this approach, by treating in a different way different portions of the thresholded 4-D spectrum. Let us remind that the 4-D spectrum is structured according to the four

dimensions of the corresponding group, i.e. two local spatial, one local temporal, and one for the non-local similarity. In particular, it includes a 2-D surface (face) corresponding to the DC terms of the two 1-D transforms used for decorrelating the temporal and non-local dimensions of the group, and 3-D volume corresponding to the DC term of the 1-D temporal transform. Hence, the value of α can be decreased for the coefficients that do not belong to this 3-D volume, in order to attenuate the temporal flickering artifacts. Likewise, the portion of spectrum in the 2-D surface can be used to characterize the group content as proposed in [45], for example by using lower values of α on flat regions to avoid noise accentuation.

We introduce the sharpening operator in the first stage (hard thresholding) only, as this guarantees excellent subjective results, and we address to future work the application of alpha-rooting during Wiener filtering.

VI. EXPERIMENTS

In this section we present the experimental results obtained with a C/MATLAB implementation of the V-BM4D algorithm. The filtering performance is measured using the PSNR, computed on the whole processed video as

$$\text{PSNR}(\hat{y}, y) = 10 \log_{10} \left(\frac{255^2 |X| |T|}{\sum_{(\mathbf{x}, t) \in X \times T} (y((\mathbf{x}, t)) - \hat{y}(\mathbf{x}, t))^2} \right), \quad (14)$$

where $|X|$ and $|T|$ stand for the cardinality of X and T , respectively. Additionally, we measure the performance of V-BM4D by means of the MOVIE index [46], a recently introduced video quality assessment (VQA) metric that is expected to be closer to the human visual judgement than the PSNR, because it concurrently evaluates space, time and jointly space-time video quality.

The transforms employed in the collaborative filtering are similar to those in [10], [11]: $\mathcal{T}_{4D}^{\text{ht}}$ (used in the hard-thresholding stage) is a 4-D separable composition of 1-D biorthogonal wavelet in both spatial dimensions, 1-D DCT in the temporal dimension, and 1-D Haar wavelet in the fourth (grouping) dimension while, $\mathcal{T}_{4D}^{\text{wie}}$ (used in the Wiener-filtering stage) differs from $\mathcal{T}_{4D}^{\text{ht}}$ as in the spatial dimension it performs a 2-D DCT. Note that, because of the Haar transform, the cardinality M of each group is set to a power of 2. To reduce the complexity of the grouping phase, we restrict the search of similar volumes within a $N_G \times N_G$ neighborhood centered around the coordinates of the reference volume, and we introduce a step of $N_{\text{step}} \in \mathbb{N}$ pixels in both horizontal and vertical directions between each reference volume. Although we set $N_{\text{step}} > 1$, we have to compute beforehand the trajectory of every possible volume in the video, since each volume is a potential candidate element of every group. Table I provides a complete overview of the parameters setting in V-BM4D.

The remaining part of this section presents the results of experiments concerning grayscale Denoising (Section VI-A), Deblocking (Section VI-B), Enhancement (Section VI-C), and Color Filtering (Section VI-D).

TABLE I

PARAMETER SETTINGS OF V-BM4D FOR THE FIRST (HARD-THRESHOLDING) AND THE SECOND (WIENER-FILTERING) STAGE. IN THE HARD-THRESHOLDING STAGE, THE THREE PARAMETERS γ_d , τ_{TRAJ} , AND τ_{MATCH} VARY ACCORDING TO THE NOISE STANDARD DEVIATION.

Stage	N	N_S	N_G	h	M	λ_{4D}	γ_p	γ_w	σ_w	N_{step}	γ_d	τ_{traj}	τ_{match}
Hard thr.	8	11	19	4	32	2.7	0.3	0.5	1	6	$\gamma_d(\sigma)$	$\tau_{\text{traj}}(\sigma)$	$\tau_{\text{match}}(\sigma)$
Wiener filt.	7		27		8	Unused					0.005	1	13.5

TABLE II

DENOISING PERFORMANCE OF V-BM3D AND V-BM4D. THE PSNR (dB) AND MOVIE INDEX [46] (THE LOWER THE BETTER) VALUES ARE REPORTED IN THE LEFT AND RIGHT PART OF EACH CELL, RESPECTIVELY. IN ORDER TO ENHANCE THE READABILITY OF THE RESULTS, EVERY MOVIE INDEX HAS BEEN MULTIPLIED BY 10^3 . THE TEST SEQUENCES ARE CORRUPTED BY WHITE GAUSSIAN NOISE WITH DIFFERENT VALUES OF STANDARD DEVIATION σ .

σ	Video:	<i>Salesm.</i>	<i>Tennis</i>	<i>Fl. Gard.</i>	<i>Miss Am.</i>	<i>Coastg.</i>	<i>Foreman</i>	<i>Bus</i>	<i>Bicycle</i>
	Res.:	288×352	240×352	240×352	288×360	144×176	288×352	288×352	576×720
	Frames:	50	150	150	150	300	300	150	30
5	V-BM4D	41.00 0.02	39.02 0.03	37.24 0.02	42.16 0.03	39.27 0.02	40.34 0.03	38.35 0.04	41.04 0.02
	V-BM3D	40.44 0.02	38.47 0.03	36.46 0.02	41.58 0.03	38.25 0.03	39.77 0.04	37.55 0.05	40.89 0.02
10	V-BM4D	37.30 0.09	35.22 0.12	32.81 0.07	40.09 0.08	35.54 0.09	36.94 0.11	34.26 0.14	37.66 0.09
	V-BM3D	37.21 0.09	34.68 0.15	32.11 0.09	39.61 0.11	34.78 0.13	36.46 0.13	33.32 0.20	37.62 0.09
15	V-BM4D	35.25 0.24	33.04 0.34	30.34 0.14	38.85 0.17	33.41 0.19	35.03 0.21	31.87 0.32	35.61 0.19
	V-BM3D	35.44 0.21	32.63 0.37	29.81 0.18	38.64 0.20	33.00 0.25	34.64 0.24	31.05 0.45	35.67 0.17
20	V-BM4D	33.79 0.46	31.59 0.60	28.63 0.23	37.98 0.27	31.94 0.32	33.67 0.33	30.26 0.53	34.10 0.30
	V-BM3D	34.04 0.46	31.20 0.73	28.24 0.28	37.85 0.31	31.71 0.41	33.30 0.38	29.57 0.72	34.18 0.27
25	V-BM4D	32.66 0.75	30.56 0.85	27.35 0.33	37.24 0.37	30.81 0.48	32.61 0.46	29.10 0.73	32.89 0.42
	V-BM3D	32.79 0.93	30.11 1.10	27.00 0.39	37.10 0.44	30.62 0.65	32.19 0.55	28.48 1.00	32.90 0.39
30	V-BM4D	31.75 1.07	29.72 1.10	26.29 0.45	36.58 0.48	29.90 0.66	31.80 0.60	28.17 0.94	31.83 0.56
	V-BM3D	31.68 1.56	29.22 1.46	25.89 0.55	36.41 0.58	29.68 0.96	31.27 0.75	27.59 1.30	31.77 0.54
35	V-BM4D	30.99 1.41	29.04 1.33	25.40 0.59	35.98 0.59	29.17 0.88	31.11 0.74	27.39 1.15	30.92 0.72
	V-BM3D	30.72 2.36	28.56 1.85	25.16 0.70	35.87 0.74	28.92 1.36	30.56 0.98	26.91 1.61	30.85 0.73
40	V-BM4D	30.35 1.76	28.49 1.56	24.60 0.75	35.47 0.70	28.54 1.13	30.52 0.89	26.72 1.37	30.10 0.89
	V-BM3D	29.93 3.09	27.99 2.17	24.33 0.92	35.45 0.89	28.27 1.86	29.97 1.21	26.28 1.93	30.02 0.94

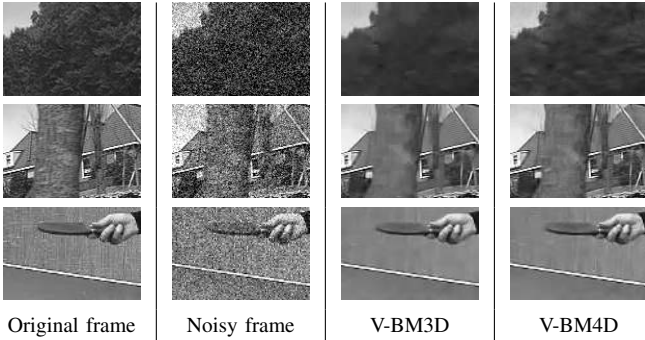


Fig. 6. From top to bottom, visual comparison of the denoising performance of V-BM4D and V-BM3D on the sequences *Bus*, *Flower Garden* and *Tennis* corrupted by white Gaussian noise with standard deviation $\sigma = 40$.

A. Grayscale Denoising

We compare the proposed filtering algorithm against V-BM3D [11], as this represents the state of the art in video denoising and we refer the reader to [11] for comparisons with other methods that are less effective than V-BM3D. Table II reports the denoising performance of V-BM3D and V-BM4D in terms of PSNR and MOVIE index. In our experiments the two algorithms are applied to a set of test sequences corrupted by white Gaussian noise with increasing standard deviation σ , which is assumed known. Observations z are obtained by synthetically adding Gaussian noise to grayscale

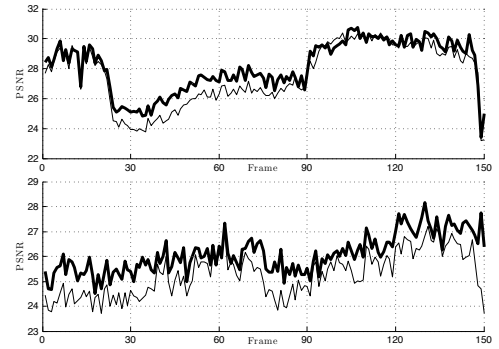


Fig. 7. Frame-by-frame PSNR (dB) output of the sequences *Tennis* (left) and *Bus* (right) corrupted by white Gaussian noise with standard deviation $\sigma = 40$ denoised by V-BM4D (thick line) and V-BM3D (thin line).

video sequences, according to (1). Further details concerning the original sequences, such as the resolution and number of frames, are reported in the header of the tables.

As one can see, V-BM4D outperforms V-BM3D in nearly all the experiments, with PSNR improvement of almost 1 dB. It is particularly interesting to observe that V-BM4D effectively handles the sequences characterized by rapid motions and frequent scene changes, especially under heavy noise, such as *Tennis*, *Flower Garden*, *Coastguard* and *Bus*. Figure 7 shows that, as soon as the sequence presents a significant change in the scene, the denoising performance decreases

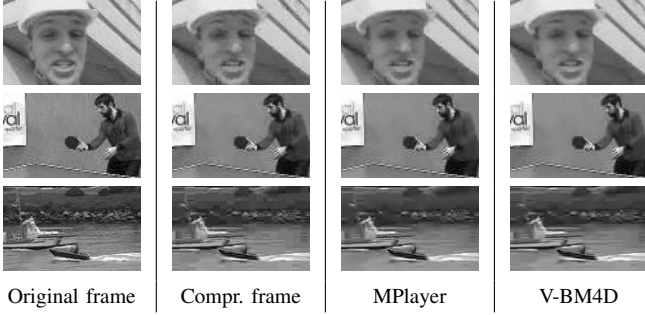


Fig. 8. Deblocking: visual comparison of V-BM4D and MPlayer on few frames. The test sequences (from top to bottom, *Foreman*, *Tennis* and *Coastguard*) have been compressed with the MPEG-4 encoder with quantization parameter $q = 25$.

significantly for both the algorithms, but, in these situations, V-BM4D requires much less frames to recover the previous PSNR values, as shown by the lower peaks at frame 90 of *Tennis*.

Finally, Figure 6 offers a visual comparison of the performance of the two algorithms. As a subjective quality assessment, V-BM4D better preserves textures, without introducing disturbing artifacts in the restored video: this is clearly visible in the tree leaves of the *Bus* sequence or in the background texture of *Tennis*. Such improvement well substantiates the considerable reduction in the MOVIE index values reported in Table II.

B. Deblocking

Table III compares, in terms of objective measurements, the V-BM4D deblocking filter against the *MPlayer accurate deblocking filter*¹, as, to the best of our knowledge, it represents one of the best deblocking algorithm. Eight sequences compressed by the MPEG-4 encoder with different values of the quantization parameter q have been considered: additional details and the bit-per-pixel rates concerning these sequences are reported in the table. Numerical results show that V-BM4D outperforms *MPlayer* in all the experiments, with improvement peaks of almost 2dB in terms of PSNR. For the sake of completeness, we also report the MOVIE index. Observe that, MOVIE often prefers the compressed observation rather than the filtered sequences, thus showing a general preference towards piecewise smooth images. However, let us observe that such results do not conform to the visual quality of the deblocked videos.

Figure 8 shows the results of V-BM4D deblocking on the *Foreman*, *Tennis* and *Coastguard* sequences, encoded at aggressive compression level ($q = 25$). The visual quality of the filtered videos has been significantly improved, since the compression artifacts, such as blocking or ghosting, have been successfully filtered without losing fine image details. In particular, we can note how the face in *Foreman*, the player and the white poster in *Tennis*, and the stone-wall in *Coastguard*, sharply emerge from their blocky counterparts, while almost-uniform areas, such as the white striped building in *Foreman*,

¹Source code and documentation can be found at <http://sourceforge.net/projects/ffds-show-tryout/> and <http://www.mplayerhq.hu/>



Fig. 9. Visual comparison of V-BM4D algorithm using different value of α . The test sequences, (*Foreman* and *Bus*), have been corrupted by white Gaussian noise with standard deviation $\sigma = 5$ (top) and $\sigma = 25$ (bottom), and have been jointly denoised and sharpened by V-BM4D.

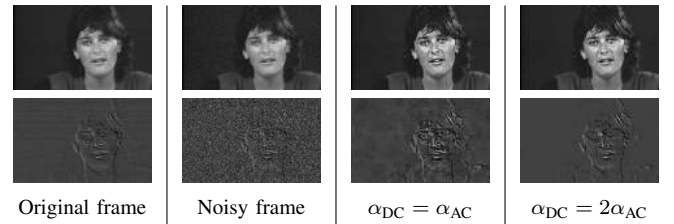


Fig. 10. Joint V-BM4D denoising, enhancement and deflickering of the sequence *Miss America* corrupted by white Gaussian noise with standard deviation $\sigma = 10$. From left to right, the bottom row shows the temporal differences between the frames presented in the top row and the preceding frames in the original, noisy, and enhanced sequences. The right-most column shows the sharpening result using different α in the temporal DC and AC coefficients of the groups spectra, thus obtaining an effective deflickering yet maintaining spatial sharpness. The images in the bottom row are all drawn with respect to the same gray colormap, which is stretched 4 times in order to improve the visualization.

or the table and the wall in *Tennis*, have been pleasingly smoothed without introducing blur.

C. Enhancement

In the enhancement experiments we use the same settings reported in Table I, testing two values of α , i.e. the parameter that controls the amount of sharpening in the alpha-rooting. Figure 9 presents the results of the V-BM4D enhancement filter applied on the *Foreman* and *Bus* sequences, corrupted by white Gaussian noise having standard deviation $\sigma \in \{5, 25\}$, and sharpened using $\alpha = 1.1$, and $\alpha = 1.25$. As the images demonstrate, the combination of V-BM4D and alpha-rooting produces satisfying results, as the fine details are effectively preserved together with a fairly good noise suppression. Such properties allowed the application of the V-BM4D enhancement filter in biomedical imaging, to facilitate the tracking of microtubules in RFP-EB3 time-lapse videomicroscopy sequences corrupted by heavy noise [2].

In particular, V-BM4D sharpens fine details, such as the tree leaves in *Bus*, and reveals barely visible information hidden in the noisy videos, as the background building of *Foreman*. The proposed enhancement filter is minimally susceptible to noise even when strong sharpening is performed (i.e., $\alpha = 1.25$), as shown by the smooth reconstruction of flat areas like the hat of *Foreman* and the bus roof of *Bus*.

TABLE III

DEBLOCKING PERFORMANCE OF V-BM4D AND MPLAYER ACCURATE DEBLOCKING FILTER. THE PSNR (dB) AND MOVIE INDEX [46] (THE LOWER THE BETTER) VALUES ARE REPORTED IN THE LEFT AND RIGHT PART OF EACH CELL, RESPECTIVELY. IN ORDER TO ENHANCE THE READABILITY OF THE RESULTS, EVERY MOVIE INDEX HAS BEEN MULTIPLIED BY 10^3 . THE PARAMETER q CONTROLS THE QUANTIZATION MATRIX OF THE MPEG-4 ENCODER AND BPP DENOTES THE AVERAGE BIT-PER-PIXEL RATE OF THE COMPRESSED VIDEO. AS A REFERENCE, WE ALSO SHOW THE PSNR AND MOVIE INDEX OF THE UNFILTERED COMPRESSED (COMPR.) VIDEOS.

q	Video:	<i>Salesm.</i>	<i>Tennis</i>	<i>Fl. Gard.</i>	<i>Miss Am.</i>	<i>Coastg.</i>	<i>Foreman</i>	<i>Bus</i>	<i>Bicycle</i>
	Res.:	288×352	240×352	240×352	288×360	144×176	288×352	288×352	576×720
	Frames:	50	150	150	150	300	300	150	30
5	bpp	0.3232	0.5323	1.4824	0.0884	0.4609	0.3005	0.7089	0.4315
	V-BM4D	35.95 0.16	34.41 0.18	33.54 0.05	39.51 0.15	34.75 0.13	36.49 0.16	35.05 0.13	38.01 0.08
	Mplayer	35.14 0.17	33.79 0.17	32.73 0.07	38.58 0.14	34.00 0.13	35.60 0.14	34.36 0.10	36.53 0.11
	Compr.	35.28 0.17	33.87 0.17	32.81 0.07	39.03 0.13	34.12 0.13	35.70 0.14	34.45 0.10	36.71 0.11
10	bpp	0.1319	0.2249	0.7288	0.0399	0.1926	0.1276	0.3285	0.2076
	V-BM4D	32.12 0.87	30.39 0.83	27.93 0.26	37.30 0.48	30.75 0.50	32.91 0.49	30.69 0.43	33.54 0.36
	Mplayer	31.66 1.08	29.87 0.89	27.40 0.31	36.61 0.53	30.23 0.53	32.16 0.52	30.11 0.41	32.45 0.46
	Compr.	31.54 0.86	29.84 0.78	27.41 0.29	36.66 0.46	30.19 0.51	32.09 0.48	30.07 0.36	32.37 0.46
15	bpp	0.0865	0.1326	0.4470	0.0318	0.1184	0.0812	0.2039	0.1333
	V-BM4D	30.06 1.89	28.48 1.49	25.15 0.58	36.13 0.82	28.73 1.01	31.10 0.90	28.48 0.85	31.16 0.79
	Mplayer	29.65 2.39	28.03 1.52	24.68 0.68	35.59 0.90	28.30 1.10	30.36 0.98	27.89 0.83	30.12 0.95
	Compr.	29.48 1.78	27.97 1.39	24.67 0.63	35.41 0.81	28.18 1.03	30.27 0.90	27.83 0.71	30.00 0.98
20	bpp	0.0661	0.0943	0.3058	0.0280	0.0852	0.0625	0.1453	0.0985
	V-BM4D	28.66 3.03	27.24 2.07	23.34 0.95	35.02 1.21	27.42 1.73	29.85 1.38	26.96 1.38	29.52 1.26
	Mplayer	28.31 3.76	26.82 2.12	22.90 1.12	32.93 1.58	27.04 1.96	29.12 1.55	26.42 1.42	28.60 1.56
	Compr.	28.11 2.71	26.76 1.93	22.88 1.02	34.21 1.21	26.90 1.73	29.03 1.37	26.35 1.16	28.43 1.58
25	bpp	0.0546	0.0710	0.2225	0.0257	0.0679	0.0523	0.1121	0.0846
	V-BM4D	27.63 4.19	26.34 2.55	22.07 1.38	34.31 1.54	26.47 2.53	29.01 1.87	25.93 1.96	28.32 1.78
	Mplayer	27.30 5.09	25.96 2.57	21.63 1.64	33.66 1.70	26.11 2.95	28.25 2.13	25.38 2.04	27.35 2.18
	Compr.	27.07 3.63	25.85 2.38	21.62 1.49	33.45 1.57	25.98 2.45	28.10 1.86	25.27 1.66	27.22 2.20
30	bpp	0.0477	0.0604	0.1697	0.0244	0.0584	0.0480	0.0921	0.0676
	V-BM4D	26.84 5.38	25.59 2.99	21.08 1.86	33.25 1.90	25.72 3.53	28.30 2.33	25.06 2.57	27.40 2.34
	Mplayer	26.51 6.31	25.26 3.02	20.65 2.24	32.80 2.08	25.38 4.20	27.57 2.68	24.55 2.70	26.54 2.88
	Compr.	26.28 4.59	25.11 2.77	20.64 1.99	32.39 1.97	25.25 3.31	27.37 2.31	24.41 2.19	26.35 2.88

As explained in Section V, the spectrum coefficients of the group can be treated differently along the temporal dimension to attenuate video temporal artifacts such as flickering. In Figure 10 we show the enhancement results of V-BM4D applied to the video *Miss America* corrupted by white Gaussian noise with $\sigma = 10$ using two settings for the sharpening parameter α . In the former experiment a fixed $\alpha_{\{DC,AC\}} = 1.25$ is used to sharpen the whole spectrum of the groups, while in the latter different values of α are used in the temporal DC and AC planes. In particular, the temporal DC coefficients are sharpened using $\alpha_{DC} = 1.25$, and the temporal AC are sharpened using the halved value $\alpha_{AC} = 0.625$. By using different values of α , V-BM4D significantly attenuates the flickering artifacts without compromising the effectiveness of neither the sharpening nor the denoising. In Figure 10, the flickering artifacts of the non-uniform intensities within the temporal difference in the background are clearly visible when the sequence is processed using $\alpha_{DC} = \alpha_{AC}$. In contrast, the sequence processed using a modified values of α exhibits a better temporal consistency as demonstrated by the smooth background in the temporal difference, yet maintaining excellent enhancement and noise reduction properties.

TABLE IV

COLOR DENOISING PERFORMANCE OF V-BM3D AND V-BM4D IN TERMS OF PSNR (dB) AND MOVIE INDEX [46] (THE LOWER THE BETTER) VALUES ARE REPORTED IN THE LEFT AND RIGHT PART OF EACH CELL, RESPECTIVELY. IN ORDER TO ENHANCE THE READABILITY OF THE RESULTS, EVERY MOVIE INDEX HAS BEEN MULTIPLIED BY 10^3 . THE TEST SEQUENCES ARE CORRUPTED BY WHITE GAUSSIAN NOISE WITH DIFFERENT VALUES OF STANDARD DEVIATION σ .

σ	Video:	<i>Tennis</i>	<i>Coastg.</i>	<i>Foreman</i>	<i>Bus</i>
	Res.:	240×352	144×176	288×352	288×352
	Frames:	150	300	300	150
5	V-BM4D	39.98 0.01	41.13 0.01	41.38 0.01	40.21 0.01
	V-BM3D	39.45 0.01	40.18 0.01	40.56 0.01	39.07 0.01
10	V-BM4D	36.42 0.04	37.28 0.03	37.92 0.05	36.23 0.05
	V-BM3D	36.04 0.04	36.82 0.03	37.52 0.04	34.96 0.07
20	V-BM4D	32.88 0.17	33.61 0.13	34.62 0.15	32.27 0.20
	V-BM3D	32.54 0.18	33.39 0.14	34.49 0.16	31.03 0.32
40	V-BM4D	29.52 0.70	30.00 0.42	31.30 0.44	28.32 0.70
	V-BM3D	29.20 0.82	29.99 0.63	31.17 0.56	27.34 1.32

D. Color Filtering

The proposed V-BM4D algorithm can be extended to color filtering using the same approach of the Color-BM3D image denoising algorithm [10], [21]. We consider the denoising of noisy color videos, such as a RGB videos, having each

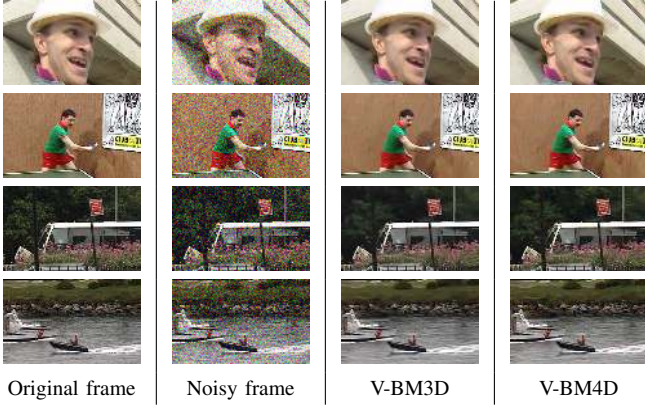


Fig. 11. Comparison of V-BM3D and V-BM4D color denoising performances. The test sequences (from top to bottom, *Foreman*, *Tennis*, *Bus* and *Coastguard*) have been corrupted by white Gaussian noise with standard deviation $\sigma = 40$.

channel independently corrupted by white Gaussian noise with variance σ^2 .

The algorithm proceeds as follows. At first, the RGB noisy video is transformed to a luminance-chrominance color space, then, both the motion estimation and the grouping are computed from the luminance channel only, as this usually has the highest SNR and carries most of the significant information. In fact, image structures do not typically vary among different channels, and the results of the motion estimation and the grouping on the luminance can be directly reused in the two chrominance channels as well. Once the groups are formed, each channel undergoes the collaborative filtering and aggregation independently, and three individual estimates are produced. Eventually, the final RGB estimate is produced by inverting the color space transformation. Such approach is a reasonable tradeoff between the achieved denoising quality and the required computational complexity. Figure 11 compares the denoising performances of V-BM4D against the state-of-the-art V-BM3D filter, on the color sequences *Foreman*, *Tennis*, *Bus*, and *Coastguard*, corrupted by white Gaussian noise having standard deviation $\sigma = 40$. As a subjective assessment, V-BM4D better preserves fine details, such as the face and the background building in *Foreman*, the background texture in *Tennis*, the leaves in *Bus* and the grass in *Coastguard*. From an objective point of view, as reported in Table IV, V-BM4D performs better than V-BM3D in every experiment, with PSNR gains of up to 1.5dB. The MOVIE index confirms the superior performances of V-BM4D, especially when the observations are corrupted with high level of noise.

VII. COMPLEXITY

In our analysis the complexity of the algorithm is measured through the number of basic arithmetic operations performed; other factors that may also influence the execution time, such as the number of memory accesses or memory consumption, have not been considered.

Each run of V-BM4D involves the execution of the hard-thresholding stage (whose complexity is $\mathcal{C}_{V-BM4D}^{\text{ht}}$), of the Wiener-filtering stage (whose complexity is $\mathcal{C}_{V-BM4D}^{\text{wie}}$), and two runs of the motion estimation algorithm (whose complexity is

TABLE V
SUMMARY OF THE PARAMETERS INVOLVED IN THE COMPLEXITY ANALYSIS.

Parameter	Notes
T	Total number of frames in the video.
n	Number of pixels per frame (i.e. $\#X$).
N	Size of the 2-D square blocks.
h	Temporal extent of the volumes in V-BM4D, size of the temporal search window in V-BM3D, corresponding to $(2N_{FR} + 1)$ in [11].
N_S	Size of the motion estimation window.
M	Size of the groups, that is the number of grouped volumes in V-BM4D or the number of grouped blocks in V-BM3D.
N_G	Size of the window used in the grouping.
N_{step}	Processing step (refer to Section VI for further details).
$\mathcal{C}_{(m,p,n)}$	Numeric operations required by a multiplication between matrices of size $m \times p$ and $p \times n$ (i.e. the cost of a linear transformation).

\mathcal{C}_{CT}). Hence, the V-BM4D overall complexity is:

$$\mathcal{C}_{V-BM4D} = 2\mathcal{C}_{CT} + \mathcal{C}_{V-BM4D}^{\text{ht}} + \mathcal{C}_{V-BM4D}^{\text{wie}}. \quad (15)$$

Differently, V-BM3D does not require any motion estimation, and thus its complexity (\mathcal{C}_{V-BM3D}) is given by the sum of the complexity of its hard-thresholding ($\mathcal{C}_{V-BM3D}^{\text{ht}}$) and Wiener-filtering ($\mathcal{C}_{V-BM3D}^{\text{wie}}$) stages:

$$\mathcal{C}_{V-BM3D} = \mathcal{C}_{V-BM3D}^{\text{ht}} + \mathcal{C}_{V-BM3D}^{\text{wie}}. \quad (16)$$

Table V shows a comprehensive summary of the parameters involved in the complexity analysis, as well as a brief description of their role in the algorithm. To provide a fair comparison, we assume that in V-BM4D the number of blocks in any spatiotemporal volume (referred as \bar{h}) coincides with the size of temporal search window N_{FR} in V-BM3D; similarly, we assume that the number of grouped volumes in V-BM4D (referred to as M) corresponds to the number of grouped blocks in V-BM3D.

A. Computation of the Trajectory

The computation of the trajectory requires searching for the most similar block within an adaptive search window of size $N_S \times N_S$ once for each of the preceding and following frames, i.e. $\bar{h} - 1$ times. Computing the ℓ^2 distance between a pair of blocks consists in $3N^2$ operations, as it requires two additions and one multiplication for each of the corresponding pixel. Since a trajectory is constructed for each pixel in every frame of the video, the total cost is

$$\mathcal{C}_{CT} = nT(\bar{h} - 1)N_S^2(3N^2). \quad (17)$$

B. Hard-Thresholding Stage

In the hard-thresholding stage, for each processed block according to N_{step} , at most M similar volumes are first extracted within a search window of size $N_G \times N_G$, then stacked together in a group, and finally transformed by a separable 4-D transform. Observe that the hard-thresholding, which is performed via element-wise comparison, requires one arithmetical operation per pixel. Eventually, the basic estimate is obtained by aggregating the inverse 4-D transform of the

filtered groups. Thus, we obtain:

$$\begin{aligned} \mathcal{C}_{\text{V-BM4D}}^{\text{ht}} = & \frac{n}{N_{\text{step}}^2} T \left(\underbrace{N_G^2 3\bar{h}N^2}_{\text{Grouping}} \right. \\ & + 2 \left(\underbrace{2M\bar{h}\mathcal{C}_{(N,N,N)} + M\mathcal{C}_{(\bar{h},\bar{h},N^2)} + \mathcal{C}_{(M,M,\bar{h}N^2)}}_{\text{Forward and Inverse Transformations}} \right) \\ & \left. + \underbrace{M\bar{h}N^2}_{\text{Thresholding}} + \underbrace{M\bar{h}N^2}_{\text{Aggregation}} \right), \end{aligned} \quad (18)$$

where the symbol $\mathcal{C}_{(\cdot,\cdot,\cdot)}$ stands for the cost of a matrix multiplication operation, as explained in Table V, and the factor 3 in the grouping complexity is due to the computation of the ℓ^2 distance between two 3-D volumes of size $N \times N \times \bar{h}$. The cost of the is the sum of four matrix multiplications, one for each dimension of the group, as this is linear and separable

In V-BM3D, the grouping is accomplished by predictive-search block-matching [11]: briefly it performs a full-search within a $N_G \times N_G$ window in the first frame to extract the N_B best-matching blocks, then, in the following \bar{h} frames, it inductively searches for other N_B best-matching blocks within windows of size $N_{PR} \times N_{PR}$ (with $N_{PR} \ll N_G$) centered at the position of the previous N_B blocks. Furthermore, since the fourth dimension is missing, the algorithm performs a 3-D transform of the M blocks of each group. The complexity of this stage is:

$$\begin{aligned} \mathcal{C}_{\text{V-BM3D}}^{\text{ht}} = & \frac{n}{N_{\text{step}}^2} T \left(\underbrace{(N_G^2 + N_B \bar{h} N_{PR}^2) 3N^2}_{\text{Grouping}} \right. \\ & + 2 \left(\underbrace{2M\mathcal{C}_{(N,N,N)} + \mathcal{C}_{(M,M,N^2)}}_{\text{Forward and inverse transformations}} \right) \\ & \left. + \underbrace{MN^2}_{\text{Thresholding}} + \underbrace{MN^2}_{\text{Aggregation}} \right). \end{aligned} \quad (19)$$

C. Wiener-filtering Stage

The complexity of the Wiener-filtering stage can be expressed as that of hard-thresholding stage in (18), with the exception that the transformation involves two groups having equal size, and that the coefficients shrinkage (performed via element-wise multiplication) involves the computation of a set of weights, which requires 6 arithmetic operations per pixel:

$$\begin{aligned} \mathcal{C}_{\text{V-BM4D}}^{\text{wie}} = & \frac{n}{N_{\text{step}}^2} T \left(\underbrace{N_G^2 3\bar{h}N^2}_{\text{Grouping}} \right. \\ & + 4 \left(\underbrace{2M\bar{h}\mathcal{C}_{(N,N,N)} + M\mathcal{C}_{(\bar{h},\bar{h},N^2)} + \mathcal{C}_{(M,M,\bar{h}N^2)}}_{\text{Forward and Inverse Transformations}} \right) \\ & \left. + \underbrace{6M\bar{h}N^2}_{\text{Shrinkage}} + \underbrace{M\bar{h}N^2}_{\text{Aggregation}} \right). \end{aligned} \quad (20)$$

Analogously, in V-BM3D the complexity of Wiener-filtering

TABLE VI

SCALABILITY OF THE V-BM4D DENOISING ALGORITHM. THE TEST SEQUENCE IS *Tennis*, CORRUPTED BY WHITE GAUSSIAN NOISE HAVING $\sigma = 25$. THE PARAMETERS M , N_G , AND $N_{\text{STEP}} = 6$ HAVE BEEN USED IN BOTH THE HARD-THRESHOLDING AND WIENER-FILTERING STAGE. TWO DIFFERENT MOTION ESTIMATION STRATEGIES HAVE BEEN EMPLOYED: A FAST DIAMOND SEARCH [47] MODIFIED IN ORDER TO INCORPORATE THE PENALTY TERM DESCRIBED IN SECTION III-A2 INTO THE BLOCK MATCHING, AND THE ONE PROPOSED IN SECTION III-A. THE TIME REQUIRED TO FILTER A SINGLE FRAME, AND (IN PARENTHESIS) THE TIME SOLELY SPENT DURING THE MOTION ESTIMATION ARE REPORTED IN THE LAST COLUMN.

Mot. est.	M	N_G	PSNR	I / fps
Mod. [47]	1	1	29.88	3.07 (2.8)
	1	19	29.88	7.36 (2.8)
	32	19	30.17	14.57 (2.8)
Sec. III-A	1	1	30.07	22.42 (22.1)
	1	19	30.07	26.76 (22.1)
	32	19	30.32	33.99 (22.1)

stage is

$$\begin{aligned} \mathcal{C}_{\text{V-BM3D}}^{\text{wie}} = & \frac{n}{N_{\text{step}}^2} T \left(\underbrace{(N_G^2 + N_B \bar{h} N_{PR}^2) 3N^2}_{\text{Grouping}} \right. \\ & + 4 \left(\underbrace{2M\mathcal{C}_{(N,N,N)} + \mathcal{C}_{(M,M,N^2)}}_{\text{Forward and Inverse Transformations}} \right) \\ & \left. + \underbrace{6MN^2}_{\text{Shrinkage}} + \underbrace{MN^2}_{\text{Aggregation}} \right). \end{aligned} \quad (21)$$

D. Comparative Analysis

The complexities of V-BM3D and V-BM4D scale linearly with the number of processed pixels, thus both algorithms are $\mathcal{O}(n)$. However, it is worth carrying out a deeper analysis since different multiplying factors may have a remarkable impact on the final cost of the two algorithms. In this comparison we assume that V-BM3D and V-BM4D share the same parameters. In this manner, we can analyze the complexities by comparing the corresponding terms of the cost expansions (18) and (19). At first, we observe that costs of the grouping can be neglected since they are similar in both algorithms. Differently, in V-BM4D the coefficients shrinkage (in the Wiener stage) and the aggregation (in both the Wiener and hard-thresholding stages) require exactly \bar{h} times more operations than in V-BM3D. We can easily compare the complexity of the transformation, as in V-BM4D it involves the additional dimension corresponding to the spatiotemporal volumes. Therefore we can conclude that the overall cost due to the transformation is more than \bar{h} times the corresponding cost in V-BM3D. An analogous inference can be made also for the costs of the Wiener-filtering stage given in (20) and (21). In conclusion, we can state that in these conditions, V-BM4D is at least \bar{h} times computationally more demanding than V-BM3D. However, V-BM4D is also burdened by the motion-estimation step, whose cost is expanded in (17). Let us observe that this cost can be entirely eliminated when the input video is encoded with a motion-compensated algorithm, such as MPEG-4 or H.264, since the motion vectors required to build the spatiotemporal volumes can be directly extracted from the encoded video.

Table VI reports the PSNR values and the corresponding seconds per frame required by V-BM4D to process the video

Tennis (CIF resolution) on a single 3GHz core. We use different settings to quantify the computational load of the grouping and the filtering, by modifying in both stages the size of the search window N_G and the number of grouped volumes M , respectively. Then, we analyze two different motion estimation strategies, specifically the predictive search described in Section III-A and the fast diamond search algorithm presented in [47] modified to incorporate the penalty term described in Section III-A2 into the block matching. Finally, we fix $N_{\text{step}} = 6$ in both stages to keep the average frame-per-second (*fps*) count unbiased. All the remaining V-BM4D parameters are set as in Table I. The speed-ups induced by the fast motion estimation algorithm ($\sim 8x$), the smaller search window ($\sim 15x$), or the smaller group size ($\sim 2.5x$), correspond to marginal PSNR losses, thus demonstrating the good scalability properties of the proposed V-BM4D. Note that, when the nonlocality features are disabled (i.e. $M = 1$ and $N_G = 1$) the motion estimation does not need to be performed for every block in the video, because only one block every N_{step} in both spatial directions is actually processed during the filtering. Thus, by skipping the motion estimation of the useless blocks, it is possible to achieve an additional speed-up of $\sim 12x$ that allows V-BM4D to process nearly 4 *fps* without affecting the final reconstruction quality.

VIII. DISCUSSION

As anticipated in the introduction, a severe limitation of V-BM3D lies in the grouping step, because it does not distinguish between the nonlocal and temporal correlation within the data. The improved effectiveness of V-BM4D indicates the importance of separately treating different types of correlation, and of explicitly accounting the motion information. In what follows we analyze how the PSNR provided by the two algorithms change when a temporal-based or nonlocal-based grouping is encouraged by varying the parameters that control the grouping strategy (both in the hard-thresholding and Wiener-filtering stage), i.e. (M, h) in V-BM4D and (N_B, N_{FR}) in V-BM3D. In these experiments we consider two videos: *Salesman* and *Tennis*, being representative of a static and a dynamic sequence, respectively.

We recall that for a given pair (M, h) V-BM4D builds volumes having temporal extent up to $2h + 1$ and stacks up to M of such volumes in the grouping step. In this analysis, we consider the pairs $(M, h) = (1, 7)$, which yields groups composed of a single volume having temporal extent 15, and $(M, h) = (16, 0)$, which yields groups composed of 16 volumes of extent having temporal extent 1. These settings correspond to a temporal-based grouping strategy in the former case, and to a nonlocal-based grouping strategy in the latter. The results reported in Table VII show that, although the temporal-based groups have a smaller number of blocks than the nonlocal-based groups, they yield a PSNR improvement of about 17% in *Salesman* and 13% in *Tennis* with respect to the basic configuration $(M, h) = (1, 0)$. In contrast, the PSNR improvement induced by nonlocal-based groups is only about 4% in *Salesman* and 3% in *Tennis*. Note that the size of the groups in V-BM4D can be reduced down to one, somehow

resembling V-BM3D, without suffering from a substantial loss in terms of restoration quality. As a matter of fact, the PSNR values shown in Table VII when $M = 1$ are only less than 0.2dB worse than the corresponding results reported in Table II, obtained using bigger values of M . Interestingly, the sequence *Salesman* shows a regular loss in performance for every $h \geq 3$ as the dimension of the groups M increases, thus manifesting that in stationary videos the nonlocality actually worsens the correlation properties of the groups.

To reproduce the nonlocal-based grouping strategy in V-BM3D, we increase the parameter N_B , controlling the number of self-similar blocks to be followed in the adjacent frames, and further we set $d_s = 0$ to give no preference towards blocks belonging to different frames (i.e. blocks having the same coordinates of the reference one [11]). Additionally we fix the maximum size of the groups to $N_2 = 16$, so that bigger groups can be formed as N_{FR} and/or N_B increase. We stress that the group composition in V-BM3D is not known when $N_B \times N_{FR} > N_2$, since the number of potential block candidates is greater than the maximum size of the group, and such candidates are unpredictably extracted from both the nonlocal and temporal dimension. Figure 12 illustrates the V-BM3D denoising performance. Similarly to V-BM4D, the graph shows a consistent PSNR improvement along the temporal dimension (i.e. as N_{FR} increases), and an almost regular loss along the nonlocal dimension (i.e. as N_B becomes larger).

This analysis empirically demonstrates that, 1) in our framework, the nonlocal spatial correlation within the data does not dramatically affect the global PSNR of the restored video, although it becomes crucial in sequences in which the temporal correlation can not be exploited (e.g., having frequent occlusions and scene changes), and 2) a grouping based only on temporal-correlated data always guarantees, both in V-BM4D and V-BM3D, higher performance than a grouping that only exploits nonlocal spatial similarity. Additionally, if the volumes are composed by blocks having the same spatial coordinate (i.e. zero motion assumption, or equivalently $\gamma_d = \infty$), the denoising quality significantly decreases: in the case of *Flower Garden* and $\sigma = 25$, the PSNR loss is ~ 2.5 dB.

IX. CONCLUSIONS

Experiments show that V-BM4D outperforms V-BM3D both in terms of objective (denoising) performance (PSNR, MOVIE index), and of visual appearance (as shown in Figure 6 and 11), thus achieving state-of-the-art results in video denoising. In particular, V-BM4D can restore fine image details much better than V-BM3D, even in sequences corrupted by heavy noise ($\sigma = 40$): this difference is clearly visible in the processed frames shown in Figure 6. However, the computational complexity of V-BM4D is obviously higher than V-BM3D, because of the motion-estimation step and the need to process higher-dimensional data. Our analysis of the V-BM4D and V-BM3D frameworks highlights that the temporal correlation is a key element in video denoising, and that it represents an effective prior that has to be exploited when designing nonlocal video restoration algorithms. Thus, V-BM4D can

TABLE VII

PSNR (dB) OUTPUTS OF V-BM4D TUNED WITH DIFFERENT SPACE (M) AND TIME (h) PARAMETERS COMBINATIONS. RECALL THAT THE TEMPORAL EXTENT IS DEFINED AS $2h + 1$. THE TEST SEQUENCES *Salesman* AND *Tennis* HAVE BEEN CORRUPTED BY WHITE GAUSSIAN NOISE WITH STANDARD DEVIATION $\sigma = 20$.

M	Video	h								
		0	1	2	3	4	5	6	7	8
1	<i>Salesm.</i>	29.22	32.22	33.12	33.54	33.78	33.93	34.03	34.10	34.16
	<i>Tennis</i>	28.04	30.38	31.04	31.33	31.48	31.56	31.61	31.64	31.65
2	<i>Salesm.</i>	29.70	32.19	32.90	33.20	33.37	33.45	33.50	33.52	33.52
	<i>Tennis</i>	28.42	30.54	31.15	31.42	31.55	31.62	31.65	31.67	31.67
4	<i>Salesm.</i>	30.08	32.32	32.92	33.14	33.22	33.24	33.22	33.18	33.13
	<i>Tennis</i>	28.63	30.62	31.18	31.42	31.52	31.56	31.57	31.56	31.53
8	<i>Salesm.</i>	30.35	32.51	33.11	33.36	33.46	33.49	33.48	33.45	33.40
	<i>Tennis</i>	28.74	30.65	31.21	31.44	31.55	31.60	31.61	31.60	31.57
16	<i>Salesm.</i>	30.47	32.65	33.29	33.57	33.72	33.79	33.82	33.81	33.80
	<i>Tennis</i>	28.78	30.66	31.21	31.45	31.56	31.63	31.65	31.66	31.65

TABLE VIII

PSNR (dB) OUTPUTS OF V-BM3D TUNED WITH DIFFERENT SPACE (N_B) AND TIME (N_{FR}) PARAMETERS COMBINATIONS. THE SIZE OF THE 3-D GROUPS HAS BEEN SET TO $N_2 = 16$ IN BOTH WIENER AND HARD-THRESHOLDING STAGES; ADDITIONALLY WE SET THE DISTANCE PENALTY TO $d_s = 0$. THE TEST SEQUENCES *Salesman* AND *Tennis* HAVE BEEN CORRUPTED BY WHITE GAUSSIAN NOISE WITH STANDARD DEVIATION $\sigma = 20$.

N_B	Video	N_{FR}								
		1	3	5	7	9	11	13	15	17
1	<i>Salesm.</i>	29.21	30.83	32.43	32.39	33.43	33.46	33.48	33.46	33.96
	<i>Tennis</i>	27.89	29.29	30.42	30.40	30.93	30.94	30.94	30.93	31.04
3	<i>Salesm.</i>	29.50	32.06	32.53	32.99	33.24	33.37	33.51	33.64	33.75
	<i>Tennis</i>	28.13	29.78	30.29	30.39	30.61	30.70	30.79	30.87	30.96
7	<i>Salesm.</i>	29.84	31.90	32.43	32.78	33.04	33.20	33.36	33.50	33.61
	<i>Tennis</i>	28.31	29.64	30.07	30.27	30.51	30.62	30.72	30.82	30.91
11	<i>Salesm.</i>	30.15	31.83	32.39	32.75	33.02	33.18	33.34	33.49	33.60
	<i>Tennis</i>	28.45	29.58	30.03	30.25	30.50	30.61	30.71	30.81	30.90
15	<i>Salesm.</i>	30.15	31.81	32.38	32.75	33.02	33.18	33.34	33.48	33.59
	<i>Tennis</i>	28.45	29.56	30.02	30.25	30.50	30.60	30.71	30.81	30.90

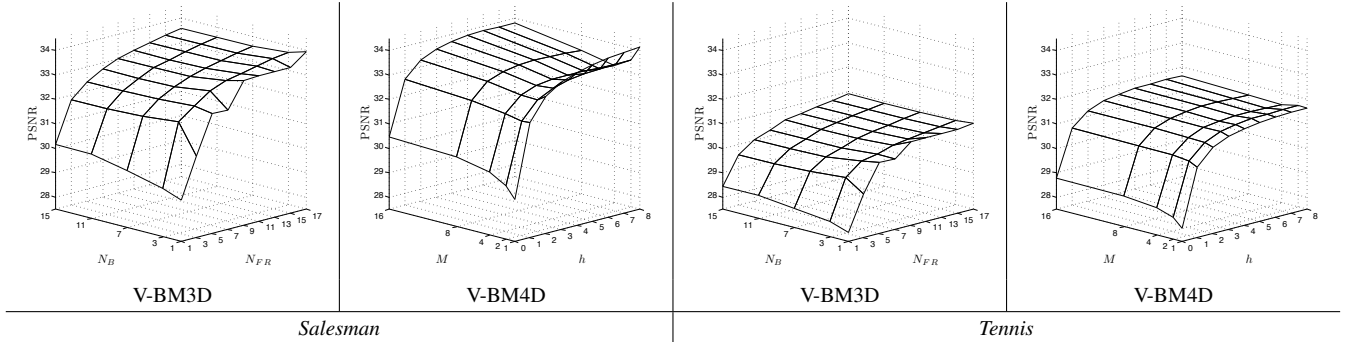


Fig. 12. PSNR (dB) surface plot of the V-BM4D and V-BM3D restoration performance for the sequence *Salesman* and *Tennis* reported in Table VII and Table VIII.

be a viable alternative to V-BM3D especially in applications where the highest restoration quality is paramount or when the separation of the four dimensions is essential.

V-BM4D can be also used as a joint denoising and sharpening filter, as well as a deblocking filter providing excellent performance on both objective and subjective visual quality. Additionally, by exploiting the separability of the 4-D transform, spatiotemporal artifacts (such as flickering) can be alleviated by acting differently on different transform coefficients. Furthermore, we remark that V-BM4D can be

extended to color data filtering in each of its applications, namely denoising, deblocking and sharpening.

REFERENCES

- [1] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising using separable 4-D nonlocal spatiotemporal transforms," in *SPIE Electronic Imaging*, Jan. 2011.
- [2] M. Maggioni, R. Mysore, E. Coffey, and A. Foi, "Four-dimensional collaborative denoising and enhancement of timelapse imaging of mCherry-EB3 in hippocampal neuron growth cones," in *BioPhotonics and Imaging Conference (BioPIC)*, Oct. 2010.

- [3] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," *IEEE Trans. on Image Proc.*, vol. 18, no. 1, pp. 27–35, 2009.
- [4] M. Ghoniem, Y. Chahir, and A. Elmoataz, "Nonlocal video denoising, simplification and inpainting using discrete regularization on graphs," *Signal Processing*, vol. 90, no. 8, pp. 2445–2455, 2010, special Section on Processing and Analysis of High-Dimensional Masses of Image and Signal Data.
- [5] J. S. De Bonet, "Noise reduction through detection of signal redundancy," Rethinking Artificial Intelligence, MIT AI Lab, Tech. Rep., 1997.
- [6] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola, "From local kernel to nonlocal multiple-model image denoising," *International Journal of Computer Vision*, vol. 86, no. 1, pp. 1–32, 2010.
- [7] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
- [8] A. Buades, B. Coll, and J.-M. Morel, "Nonlocal image and movie denoising," *Int. Journal of Computer Vision*, vol. 76, no. 2, pp. 123–139, 2008.
- [9] X. Li and Y. Zheng, "Patch-based video processing: a variational bayesian approach," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 29, pp. 27–40, Jan. 2009.
- [10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, August 2007.
- [11] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," in *European Signal Processing Conference (EUSIPCO)*, Poznan, Poland, Sep. 2007.
- [12] G. Boracchi and A. Foi, "Multiframe raw-data denoising based on block-matching and 3-D filtering for low-light imaging and stabilization," in *Int. Workshop on Local and Non-Local Approx. in Image Proc. (LNLA)*, 2008.
- [13] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Joint image sharpening and denoising by 3D transform-domain collaborative filtering," in *Int. TICSP Workshop Spectral Meth. Multirate Signal Process. (SMMSP)*, 2007.
- [14] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, "Image and video super-resolution via spatially adaptive block-matching filtering," in *Int. Workshop on Local and Non-Local Approx. in Image Process. (LNLA)*, August 2008.
- [15] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, "Image up-sampling via spatially adaptive block-matching filtering," in *European Signal Processing Conference, (EUSIPCO)*, 2008.
- [16] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, *Spatially adaptive filtering as regularization in inverse imaging: compressive sensing, upsampling, and super-resolution*, in *Super-Resolution Imaging*. CRC Press/Taylor & Francis, Sep. 2010.
- [17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image restoration by sparse 3D transform-domain collaborative filtering," in *SPIE Electronic Imaging*, vol. 6812, no. 6812-1D, Jan. 2008.
- [18] H.-M. Hang, Y.-M. Chou, and S.-C. Cheng, "Motion estimation for video coding standards," *Journal of VLSI Signal Processing Systems*, vol. 17, no. 2/3, pp. 113–136, 1997.
- [19] R. Megret and D. Dementhon, "A survey of spatio-temporal grouping techniques," Tech. Rep., 2002.
- [20] A. Basharat, Y. Zhai, and M. Shah, "Content based video matching using spatiotemporal volumes," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 360–377, 2008.
- [21] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," in *Int. Conf. Image Process. (ICIP)*, Sep. 2007.
- [22] H. Oktem, V. Katkovnik, K. Egiazarin, and J. Astola, "Local adaptive transform based image denoising with varying window size," in *Int. Conf. on Image Proc.*, vol. 1, 2001, pp. 273–276.
- [23] O. Guleryuz, "Weighted overcomplete denoising," in *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, vol. 2, Nov. 2003, pp. 1992–1996.
- [24] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer J.*, vol. 7, pp. 308–313, 1965.
- [25] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM J. of Opt.*, vol. 9, pp. 112–147, 1998.
- [26] "The MPEG-4 video standard verification model," pp. 142–154, 2001.
- [27] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.
- [28] T. Chen, H. Wu, and B. Qiu, "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 5, pp. 594–602, May 2001.
- [29] B. Gunturk, Y. Altunbasak, and R. Mersereau, "Multiframe blocking-artifact reduction for transform-coded video," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 4, pp. 276–282, April 2002.
- [30] J. Chou, M. Crouse, and K. Ramchandran, "A simple algorithm for removing blocking artifacts in block-transform coded images," pp. 33–35, Feb. 1998.
- [31] S.-W. Hong, Y.-H. Chan, and W.-C. Siu, "Subband adaptive regularization method for removing blocking effect," vol. 2, p. 2523, Oct. 1995.
- [32] A. Wong and W. Bishop, "Deblocking of block-transform compressed images using phase-adaptive shifted thresholding," in *IEEE Int. Symp. on Multimedia (ISM)*, Dec. 2008, pp. 97–103.
- [33] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image Process.*, vol. 16, no. 5, May 2007.
- [34] Z. Wang, A. Bovik, and B. Evan, "Blind measurement of blocking artifacts in images," in *Int. Conf. on Image Proc. (ICIP)*, vol. 3, 2000, pp. 981–984.
- [35] S. Agaian, B. Silver, and K. Panetta, "Transform coefficient histogram-based image enhancement algorithms using contrast entropy," *IEEE Trans. on Image Proc.*, vol. 16, no. 3, pp. 741–758, Mar. 2007.
- [36] G. Ramponi, "Polynomial and rational operators for image processing and analysis," pp. 203–223, 2001.
- [37] G. Ramponi, N. Strobel, S. K. Mitra, and T. Yu, "Nonlinear unsharp masking methods for image contrast enhancement," *Journal of Electronic Imaging*, vol. 5, pp. 353–366, July 1996.
- [38] A. Polesel, G. Ramponi, and V. Mathews, "Image enhancement via adaptive unsharp masking," *IEEE Trans. on Image Proc.*, vol. 9, no. 3, pp. 505–510, Mar. 2000.
- [39] F. Russo, "An image enhancement technique combining sharpening and noise reduction," in *IEEE Instrumentation and Measurement Technology Conference (IMTC)*, vol. 3, 2001, pp. 1921–1924.
- [40] T. Aysal and K. Barner, "Quadratic weighted median filters for edge enhancement of noisy images," *IEEE Trans. on Image Proc.*, vol. 15, no. 11, pp. 3294–3310, Nov. 2006.
- [41] J. Fischer, M. and Paredes and G. Arce, "Image sharpening using permutation weighted medians," in *Proc. X Eur. Signal Processing Conf. Tampere, Finland*, Sep. 2000.
- [42] S. Aghagholzadeh and O. K. Ersoy, "Transform image enhancement," *Optical Engineering*, vol. 31, no. 3, pp. 614–626, 1992. [Online]. Available: <http://link.aip.org/link/?JOE/31/614/1>
- [43] S. Hatami, R. Hosseini, M. Kamarei, and H. Ahmadi, "Wavelet based fingerprint image enhancement," in *IEEE Int. Symposium on Circuits and Systems (ISCAS)*, vol. 5, May 2005, pp. 4610–4613.
- [44] J. McClellan, "Artifacts in alpha-rooting of images," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, Apr. 1980, pp. 449–452.
- [45] H. Tong and A. Venetsanopoulos, "A perceptual model for JPEG applications based on block classification, texture masking, and luminance masking," in *Int. Conf. on Image Proc. (ICIP)*, vol. 3, Oct. 1998, pp. 428–432.
- [46] K. Seshadrinathan and A. C. Bovik, "Motion tuned spatio-temporal quality assessment of natural videos," *IEEE Trans. on Image Proc.*, vol. 19, no. 2, pp. 335–350, 2010.
- [47] Y. Ismail, J. McNeelly, M. Shaaban, and M. Bayoumi, "Enhanced efficient diamond search algorithm for fast block motion estimation," in *IEEE Int. Symposium on Circuits and Systems (ISCAS)*, May 2009, pp. 3198–3201.