Video denoising using separable 4-D nonlocal spatiotemporal transforms

Matteo Maggioni°, Giacomo Boracchi⋆, Alessandro Foi°, Karen Egiazarian°

°Department of Signal Processing, Tampere University of Technology, Finland;
⋆Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy

ABSTRACT

We propose a powerful video denoising 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 is leveraged to enforce sparsity and thus regularize the data. The proposed algorithm exploits the mutual similarity between 3-D spatiotemporal volumes constructed by tracking blocks along trajectories defined by the motion vectors. Mutually similar volumes are 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. Collaborative filtering is realized by transforming each group through a decorrelating 4-D separable transform and then by shrinkage and inverse transformation. In this way, collaborative filtering provides estimates for each volume stacked in the group, which are then returned and adaptively aggregated to their original position in the video. Experimental results demonstrate the effectiveness of the proposed procedure which outperforms the state of the art.

Keywords: Video denoising, nonlocal methods, adaptive transforms, motion estimation

1. INTRODUCTION

The large number of practical applications involving digital videos has motivated a significant interest in denoising solutions, and the literature contains a plethora of such algorithms (see[1, 2] for a comprehensive overview). At the moment, the most effective approach in restoring images or videos exploits the redundancy given by the *nonlocal* similarity between patches at different locations within the data.[3] Algorithms based on this approach have been proposed for various signal processing problems, and mainly for denoising.[2–12] Among these methods, we especially mention the BM3D algorithm,[7] which represents the state of the art in image denoising. BM3D 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 into the estimate of the 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 applied successfully to video denoising (V-BM3D),[8] as well as to several other applications including image and video super-resolution,[11–13] image sharpening,[10] and image deblurring.[14]

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. However, it is typically along the temporal dimension that most mutually similar blocks can be found. It is well known that motion-compensated videos[15] are extremely smooth along the temporal axis and this fact is exploited by nearly all modern video-coding techniques. As shown by the experimental analysis

This work was supported by the Academy of Finland (project no. 213462, Finnish Programme for Centres of Excellence in Research 2006-2011, project no. 118312, Finland Distinguished Professor Programme 2007-2010, and project no. 129118, Postdoctoral Researcher's Project 2009-2011).

in,[9] even when 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 is due to the tracking of motion along time or to the nonlocal spatial self-similarity within each frame. In other words, the filtering in V-BM3D is not able to distinguish between temporal versus spatial nonlocal similarity. We recognize it as a conceptual as well as practical weakness of the algorithm: as simple experiments can demonstrate, increasing the number of spatially self-similar blocks in a V-BM3D group does not lead to an improvement in the final result and instead it most often leads to a systematic degradation.

This work proposes V-BM4D, a novel video-denoising approach that, to overcome the above weaknesses, separately exploits the temporal and spatial redundancy in the video sequence. For the sake of clarity and because of space limitation, we present V-BM4D for denoising only, although it can be implemented for a variety of other video filtering applications. The core element of V-BM4D is the spatiotemporal volume, a 3-D structure formed by a sequence of blocks extracted from the noisy video following a specific trajectory (obtained, for example, by concatenating motion vectors along time).[16,17] Thus, contrary to V-BM3D, V-BM4D does not group blocks, but mutually similar spatiotemporal volumes according to a nonlocal search procedure. Hence, these groups are 4-D stacks of 3-D volumes and the collaborative filtering is then performed via a separable 4-D spatiotemporal transform. The transform takes advantage of the following three types of correlation that characterize natural video sequences:

- the local spatial correlation between pixels in each block of a volume;
- the local temporal correlations between blocks of each volume;
- the nonlocal spatial and temporal correlation between grouped volumes.

The 4-D group spectrum is thus highly sparse, which makes the shrinkage more effective than in V-BM3D and results in the superior performance of V-BM4D in terms of noise reduction.

The paper is organized as follows: Section 3 presents a formal definition of the fundamental steps of the algorithm, while Section 4 describes the implementation aspects, with particular attention to the computation of motion vectors; experiments are illustrated and discussed in Section 5.

2. OBSERVATION MODEL

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

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

where y is the original video, $\eta(\cdot,\cdot) \sim \mathcal{N}(0, \sigma^2)$ is i.i.d. 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 index t is denoted by $z(X, t)$.

3. BASIC ALGORITHM

The aim of the proposed algorithm is to provide an estimate $\hat{y}$ of the original video y from the observed data z. According to the BM3D paradigm, the V-BM4D algorithm comprises three fundamental steps, specifically grouping, collaborative filtering and aggregation. These steps are performed for every spatiotemporal volume of the video.

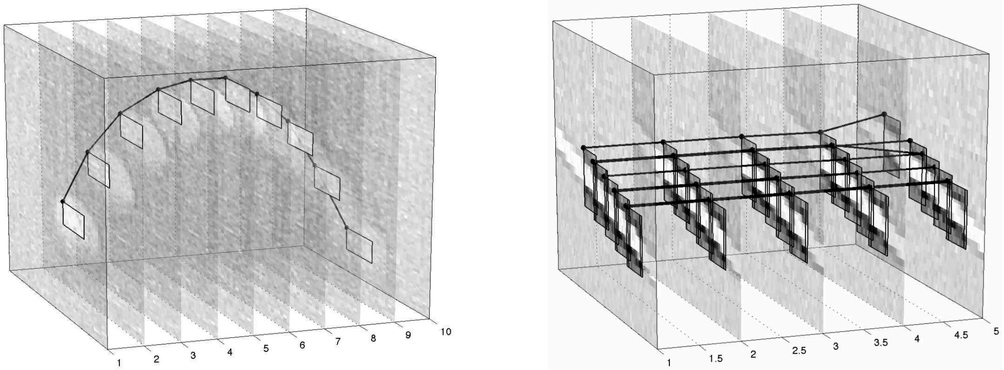


Figure 1. Illustration of trajectory and associated volume (left), and group of mutually similar volumes (right) calculated for the sequence *Tennis* corrupted by white Gaussian noise with $\sigma = 20$.

3.1 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 the 3-D sequence of blocks built following a specific trajectory along time. The trajectory associated to $(\mathbf{x}_0, t_0)$ is defined as

$$\mathrm{Traj}(\mathbf{x}_0, t_0) = \Big\{(\mathbf{x}_j, t_0 + j)\Big\}_{j=-h^-}^{h^+}, \tag{2}$$

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 $z(X, t_0+j)$, $j = -h^-, \ldots, h^+$. For the sake of simplicity, in this section it is assumed $h^- = h^+ = h$ for all $(\mathbf{x}, t) \in X \times T$ and the considerations concerning the general case are postponed in Section 4.

The trajectories can be either computed from the noisy video (as shown in Section 4.1), or, when given a coded video, 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 $\mathrm{Traj}(\mathbf{x}_0, t_0)$ is given and thus the 3-D spatiotemporal volume in $(\mathbf{x}_0, t_0)$ can be determined as

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

where the subscript z specifies that the volumes are extracted from the noisy video. The length of a volume $V_z(\mathbf{x}_i, t_i)$ is defined as

$$L_i = h^- + h^+ + 1. \tag{4}$$

3.2 Grouping

Groups are stacks of mutually similar volumes and constitute the nonlocal element of V-BM4D. Mutually similar volumes are determined with a nonlocal search procedure as in.[7] Let $\mathrm{Ind}(\mathbf{x}_0, t_0)$ be the set of indexes identifying volumes that, according to a distance operator δ^{v}, are similar to $V_z(\mathbf{x}_0, t_0)$:

$$\mathrm{Ind}(\mathbf{x}_0, t_0) = \big\{(\mathbf{x}_i, t_i) : \delta^{\mathrm{v}}(V_z(\mathbf{x}_0, t_0), V_z(\mathbf{x}_i, t_i)) < \tau_{\mathrm{match}}\big\}. \tag{5}$$

The parameter $\tau_{\mathrm{match}} > 0$ controls the minimum degree of similarity among volumes; the distance δ^{v} 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

$$G_z(\mathbf{x}_0, t_0) = \big\{V_z(\mathbf{x}_i, t_i) : (\mathbf{x}_i, t_i) \in \mathrm{Ind}(\mathbf{x}_0, t_0)\big\}. \tag{6}$$

In (6), we implicitly assume that the 3-D volumes are stacked along a fourth dimension, and hence the groups are 4-D data structures. Note that since $\delta^{\mathrm{v}}(V_z, V_z) = 0$, 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 examples of trajectories, volumes and groups.

3.3 Collaborative Filtering

In the general formulation of the grouping and collaborative-filtering approach for a d-dimensional signal,[7] 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 with 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:

$$\hat{G}_y(\mathbf{x}_0, t_0) = \mathcal{T}_{4D}^{-1}\big(\Upsilon\left(\mathcal{T}_{4D}\left(G_z(\mathbf{x}_0, t_0)\right)\right)\big), \tag{7}$$

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

$$\hat{G}_y(\mathbf{x}_0, t_0) = \big\{\hat{V}_y(\mathbf{x}_i, t_i) : (\mathbf{x}_i, t_i) \in \mathrm{Ind}(\mathbf{x}_0, t_0)\big\}, \tag{8}$$

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

3.4 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

$$\hat{y} = \frac{\sum_{(\mathbf{x}_0,t_0)\in X\times T}\left(\sum_{(\mathbf{x}_i,t_i)\in \mathrm{Ind}(\mathbf{x}_0,t_0)} w_{(\mathbf{x}_0,t_0)}\hat{V}_y(\mathbf{x}_i, t_i)\right)}{\sum_{(\mathbf{x}_0,t_0)\in X\times T}\left(\sum_{(\mathbf{x}_i,t_i)\in \mathrm{Ind}(\mathbf{x}_0,t_0)} w_{(\mathbf{x}_0,t_0)}\chi_{(\mathbf{x}_i,t_i)}\right)}, \tag{9}$$

where we assume $\hat{V}_y(\mathbf{x}_i, t_i)$ to be zero-padded outside its domain, $\chi_{(\mathbf{x}_i,t_i)} : X \times T \to \{0, 1\}$ 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. The particular choice of the aggregation weights depends on the result of shrinkage in the collaborative filtering: typically the weights are defined so that the sparser is the shrunk 4-D spectrum $\hat{G}_y(\mathbf{x}_0, t_0)$, the larger is the weight $w_{(\mathbf{x}_0,t_0)}$. In particular, the weights can be effectively defined to be inversely proportional to the total sample variance of the estimate of the corresponding groups.[7]

4. IMPLEMENTATION ASPECTS

4.1 Computation of the Trajectories

In our implementation, we construct trajectories by concatenation of motion vectors which are defined as follows.

4.1.1 Similarity criterion

Motion of a block is generally tracked by identifying the most similar block in the subsequent (and precedent) frame. However, since we deal with noisy signals, prior information about motion smoothness can be exploited to improve the tracking. In particular, provided a rough guess $\hat{\mathbf{x}}_i(t_j)$ of the future (or past) location of the block $B_z(\mathbf{x}_i, t_i)$ at the time $t_j = t_i + 1$ $(t_j = t_i - 1)$, we define the similarity between $B_z(\mathbf{x}_i, t_i)$ and $B_z(\mathbf{x}_j, t_j)$, through a penalized quadratic difference

$$\delta^{\mathrm{b}}\big(B_z(\mathbf{x}_i, t_i), B_z(\mathbf{x}_j, t_j)\big) = \frac{\left|\left|B_z(\mathbf{x}_i, t_i) - B_z(\mathbf{x}_j, t_j)\right|\right|_2^2}{N^2} + \gamma_d \left|\left|\hat{\mathbf{x}}_i(t_j) - \mathbf{x}_j\right|\right|_2, \tag{10}$$

where $\hat{\mathbf{x}}_i(t_j)$ is the predicted position of $B_z(\mathbf{x}_i, t_i)$ in the frame $z\,(X, t_j)$, and $\gamma_d \in \mathbb{R}^+$ is the penalization parameter. Whenever $\hat{\mathbf{x}}_i(t_j)$ is not available, we consider the lack of motion as the most likely condition and we set $\hat{\mathbf{x}}_i(t_j) = \mathbf{x}_i$.

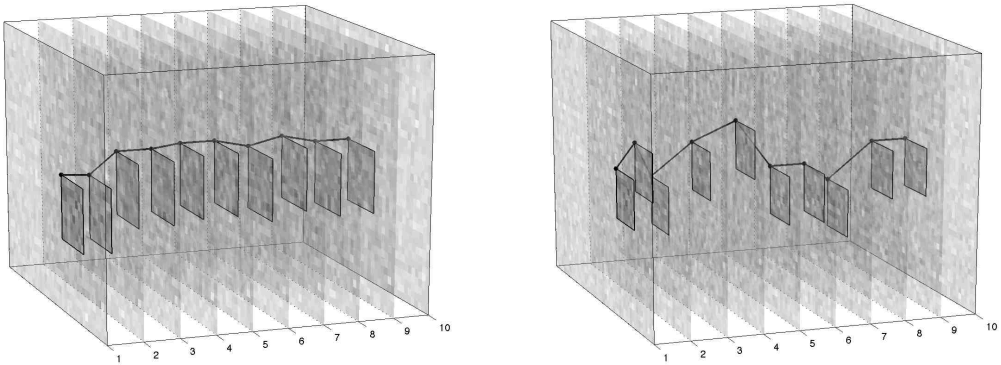


Figure 2. Effect of different penalties $\gamma_d = 0.025$ (left) and $\gamma_d = 0$ (right) on the background textures of the sequence *Tennis* corrupted by Gaussian noise with $\sigma = 20$. The initial positions at time $t = 1$ are equal in both experiments.

V-BM4D repeatedly uses the minimization of (10) to construct the trajectory (2). Formally, the motion of $B_z(\mathbf{x}_i, t_i)$ from time t_i to $t_i + 1$ is determined by the position $\mathbf{x}_j$ that minimizes (10)

$$\mathbf{x_j} = \underset{x_k \in \mathcal{N}}{\arg\min} \left\{ \delta^{\mathrm{b}}\big(B_z(\mathbf{x}_i, t_i), B_z(\mathbf{x}_k, t_i + 1)\big) < \tau_{\mathrm{traj}} \right\}, \tag{11}$$

where $\mathcal{N}$ is a restriction in the frame $z(X, t_i + 1)$ applied by an adaptive search neighborhood (further details are given in Section 4.1.3). Nevertheless, even though a minimizer for (10) can always be found, we interrupt the trajectory whenever the corresponding minimum distance δ^{b} exceeds a fixed parameter $\tau_{\mathrm{traj}} \in \mathbb{R}^+$, which determines the minimum accepted similarity along spatiotemporal volumes, to effectively deal with occlusions and changes of scene. Figure 2 illustrates trajectories estimated using different penalization parameters. Observe that the penalization term is essential when tracking blocks belonging to areas covered by homogeneous texture, in fact, as shown in the right image of Figure 2, without a position-dependent distance metric, the trajectories would be mainly determined by noise, and, for this reason, the collaborative filtering would be less effective.

4.1.2 Location prediction

As soon as the motion of a block at two consecutive spatiotemporal locations $(\mathbf{x}_{i-1}, t_i - 1)$ and $(\mathbf{x}_i, t_i)$ has been determined, we can define the motion vector (velocity) $\mathbf{v}(\mathbf{x}_i, t_i) = \mathbf{x}_{i-1} - \mathbf{x}_i$. Hence, under the assumption of smooth motion, we define the guess $\hat{\mathbf{x}}_i(t_i + 1)$ as

$$\hat{\mathbf{x}}_i(t_i + 1) = \mathbf{x}_i + \gamma_p \cdot \mathbf{v}(\mathbf{x}_i, t_i), \tag{12}$$

where $\gamma_p \in [0, 1]$ is a weighting factor of the prediction. Analogous prediction can be made for $\hat{\mathbf{x}}_{i-1}(t_i - 1)$, when we look for precedent blocks in the sequence.

4.1.3 Search neighborhood

Because of the penalty term $\gamma_d \, ||\hat{\mathbf{x}}_i(t_j) - \mathbf{x}_j||_2$, the minimizer of (10) is likely close to $\hat{\mathbf{x}}_i(t_j)$. We therefore restrict the minimization of (10) to a spatial search neighborhood $\mathcal{N}$ centered at $\hat{\mathbf{x}}_i(t_j)$. The size $N_{PR} \times N_{PR}$ of this neighborhood can be adapted based on the velocity (magnitude of motion vector) of the tracked block by setting

$$N_{PR} = N_S \cdot \left(1 - \gamma_w \cdot e^{-\frac{||\mathbf{v}(\mathbf{x}_i, t_i)||_2^2}{2 \cdot \sigma_w^2}} \right), \tag{13}$$

where N_S is the maximum size of $\mathcal{N}$, $\gamma_w \in [0, 1]$ is a scaling factor and $\sigma_w > 0$ is a tuning parameter. As the velocity 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 how fast the exponential term approaches zero, or, in other words, how permissive is the window shrinkage with respect to the velocity of the tracked block. For instance, considering the same velocity $\mathbf{v}$ for a given block and using increasing values of σ_w in (13), we would obtain smaller windows, because the decay of the function would be slower.

4.2 Sub-volume Extraction

So far, the number of frames spanned by all the trajectories has been assumed fixed. However, because of occlusions, scene changes or heavy noise, any trajectory $\text{Traj}(\mathbf{x}_i, t_i)$ can be interrupted at any time, as determined by the parameter τ_{traj}. Thus, if $\left[t_i - h_i^-, t_i + h_i^+\right]$ is the temporal extent of the trajectory $\text{Traj}(\mathbf{x}_i, t_i)$, we have that

$$0 \leq h_i^- \leq h\,, \quad 0 \leq h_i^+ \leq h, \tag{14}$$

where h denotes the maximum forward and backward extent of trajectories (and thus volumes) allowed in the algorithm.

As a result, during grouping, V-BM4D may stack together volumes having different lengths. Nevertheless, 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 of equal length. Thus, in the current implementation of grouping we consider, for each reference volume $V_z(\mathbf{x}_0, t_0)$, only volumes $V_z(\mathbf{x}_i, t_i)$ such that $t_i = t_0$, $h_i^- \geq h_0^-$ and $h_i^+ \geq h_0^+$. In this case, V-BM4D extracts from $V_z(\mathbf{x}_i, t_i)$ the sub-volume with temporal extent $[t_0 - h_0^-, t_0 + h_0^+]$, denoted as $\mathcal{E}_{L_0}\big(V_z(\mathbf{x}_i, t_i)\big)$. There are obviously many other, less restrictive, possibilities for extracting sub-volumes of length L_0 from longer volumes, however, the one we implemented aims at limiting the complexity while maintaining a high correlation within the grouped volumes.

In the grouping, the distance operator δ^{v} is the ℓ^2-norm of the difference between time-synchronous volumes normalized with respect to their lengths

$$\delta^{\text{v}}\big(V_z(\mathbf{x}_0, t_0), V_z(\mathbf{x}_i, t_i)\big) = \big|\big|V_z(\mathbf{x}_0, t_0) - \mathcal{E}_{L_0}\big(V_z(\mathbf{x}_i, t_i), t_0\big)\big|\big|_2^2 / L_0, \tag{15}$$

thus providing larger weight to the volumes belonging to groups having sparser representation in $\mathcal{T}_{4D}$ domain.

4.3 Two-Stage Implementation with Collaborative Wiener Filtering

The general procedure described in Section 3 is implemented in two cascading stages, both composed of the grouping, collaborative filtering and aggregation steps.

4.3.1 Hard-thresholding stage

In the first stage, volumes are extracted from the noisy video z, and groups are then formed using the similarity measure δ^{v}-operator (15), and the predefined threshold $\tau_{\text{match}}^{\text{ht}}$. Collaborative filtering is realized by hard thresholding in 4-D transform domain each group $G_z(\mathbf{x}, t)$:

$$\hat{G}_y^{\text{ht}}(\mathbf{x}, t) = \mathcal{T}_{4D}^{\text{ht}^{-1}}\left(\Upsilon^{\text{ht}}\left(\mathcal{T}_{4D}^{\text{ht}}\left(G_z(\mathbf{x}_0, t_0)\right)\right)\right),\ (\mathbf{x}, t) \in X \times T, \tag{16}$$

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 hard-thresholding stage, $\hat{y}^{\text{ht}}$, is obtained by aggregation of all the estimated groups $\hat{G}_y^{\text{ht}}(\mathbf{x}, t)$. The weights $w_{(\mathbf{x}_0, t_0)}^{\text{ht}}$ in the aggregation (9) are inversely proportional to the number $N_{(\mathbf{x}_0, t_0)}^{\text{ht}}$ of non-zero coefficients of the corresponding hard-thresholded group $\hat{G}_y^{\text{ht}}(\mathbf{x}_0, t_0)$:

$$w_{(\mathbf{x}_0, t_0)}^{\text{ht}} = \frac{1}{N_{(\mathbf{x}_0, t_0)}^{\text{ht}}}. \tag{17}$$

4.3.2 Wiener filtering stage

In the second stage, new trajectories $\text{Traj}_{\hat{y}^{\text{ht}}}$ are extracted from the basic estimate $\hat{y}^{\text{ht}}$, 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 set of volume indexes $\text{Ind}_{\hat{y}^{\text{ht}}}(\mathbf{x}, t)$ resulting from similarity search are used to construct two sets of 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.

Table 1. Parameter settings of V-BM4D for the first (hard-thresholding) and the second (Wiener-filtering) stage. The parameters γ_d, τ_{traj} and τ_{match} vary according to the noise, as shown in Figure 3.

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*				4	0.005	1	13.5

Collaborative filtering is hence performed using an empirical Wiener filter in $\mathcal{T}_{4D}^{\text{wie}}$ transform domain, whose shrinkage coefficients are computed from the energy of the 4-D spectrum of the basic estimate group $G_{\hat{y}^{\text{ht}}}$

$$\mathbf{W}(\mathbf{x}_0, t_0) = \frac{\left|\mathcal{T}_{4D}^{\text{wie}}\left(G_{\hat{y}^{\text{ht}}}(\mathbf{x}_0, t_0)\right)\right|^2}{\left|\mathcal{T}_{4D}^{\text{wie}}\left(G_{\hat{y}^{\text{ht}}}(\mathbf{x}_0, t_0)\right)\right|^2 + \sigma^2}, \tag{18}$$

Shrinkage is realized as element-by-element multiplication between the 4-D transform coefficients of the group $G_z(\mathbf{x}_0, t_0)$ extracted from the noisy video z and the Wiener coefficients $\mathbf{W}(\mathbf{x}_0, t_0)$. Subsequently, we obtain the group of volumes estimates by inverting the 4-D transform as

$$\hat{G}_y^{\text{wie}}(\mathbf{x}_0, t_0) = \mathcal{T}_{4D}^{\text{wie}^{-1}}\left(\mathbf{W}(\mathbf{x}_0, t_0) \cdot \mathcal{T}_{4D}^{\text{wie}}\left(G_z(\mathbf{x}_0, t_0)\right)\right). \tag{19}$$

The global final estimate $\hat{y}^{\text{wie}}$ is computed by the aggregation (9), using the weights

$$w_{(\mathbf{x}_0, t_0)}^{\text{wie}} = ||\mathbf{W}(\mathbf{x}_0, t_0)||_2^{-2}. \tag{20}$$

5. EXPERIMENTS

In this section we present the experimental results obtained with a C/MATLAB implementation of the V-BM4D algorithm, and we compare it against V-BM3D*, as it represents the state of the art in video denoising. Observations z are obtained by synthetically adding Gaussian noise to greyscale image sequences, according to (1). The denoising performance is measured using the PSNR as a global measure for the whole processed video:

$$\text{PSNR} = -10 \log_{10}\left(255^{-2}|X||T| \sum_{(\mathbf{x},t)\in X\times T}\big(y((\mathbf{x}, t)) - \hat{y}(\mathbf{x}, t)\big)^{-2}\right), \tag{21}$$

where $|X|$ and $|T|$ stand for the cardinality of X and T, respectively.

The transforms employed in the collaborative filtering are similar to those in:[7,8] in the hard-thresholding stage $\mathcal{T}_{4D}^{\text{ht}}$ 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, in the Wiener filtering stage, $\mathcal{T}_{4D}^{\text{wie}}$ uses a 2-D DCT for the spatial dimension. Note that, because of the Haar transform, the cardinality M of each group must be a power of 2. In order 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, moreover, to lighten the computational complexity of the grouping, a step of $N_{\text{step}} \in \mathbb{N}$ pixels in both horizontal and vertical directions separates each processed volume. Notwithstanding the trajectory of every possible volume in the video must be computed beforehand, because any volume is a potential candidate element of every group.

The two stages share some of the parameters such as: the search neighborhoods for the trajectory calculation N_S, the temporal extent h, the weights γ_p of (12) and γ_w, σ_w of (13), while the block size N, the grouping window N_G, the group size M, and the processing step N_{step} are different, and λ_{4D} is used in the first stage only. Observe that we restrict the volumes contained in the groups to be the largest power of 2 smaller than or equal to the minimum value between the original cardinality of the groups and M itself.

*Matlab code at http://www.cs.tut.fi/∼foi/GCF-BM3D/.

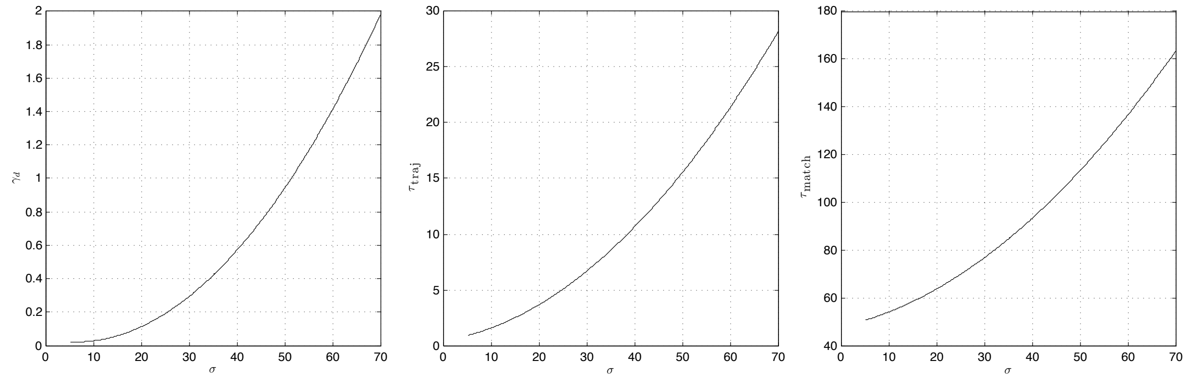
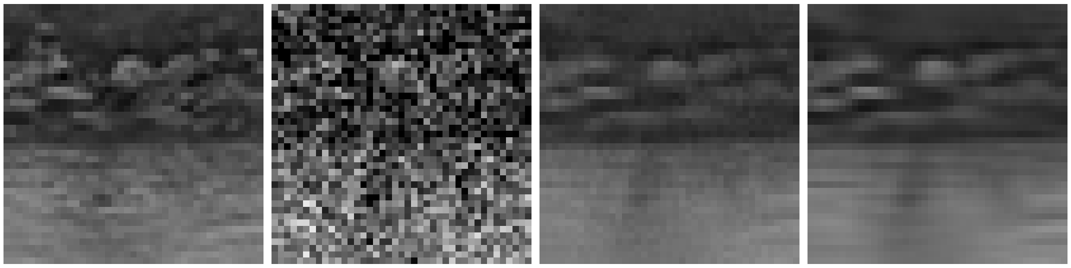


Figure 3. Parameters depending on σ in the hard-thresholding stage. The functions are the quadratic polynomials approximation of the optimum parameters obtained from the Nelder-Mead simplex direct search algorithm applied on a set of test sequences corrupted by white Gaussian noise having different values of σ. The functions are built such that their coefficients maximize the average PSNR of the test sequences along each value of σ. In particular we use *Salesman*, *Tennis*, *Flower Garden Miss America*, *Coastguard*, *Foreman*, *Bus*, and *Bicycle*.



(a) Original y

(b) Noisy z

(c) Result y^{ht} of the first stage

(d) Result y^{wie} of the second stage

Figure 4. Visual comparison of the sequence *Coastguard* corrupted by white Gaussian noise with standard deviation $\sigma = 40$, denoised after the first and second stage of V-BM4D.

The parameters involved in the motion estimation and in the grouping, that is γ_d, τ_{traj} and τ_{match}, vary with σ. Intuitively, in order to compensate the effects of the noise, the larger σ is, the larger the thresholds controlling blocks and volumes matching become. The behavior of such parameters w.r.t. σ is determined following an empirical approach. First we compute the parameters that maximize the V-BM4D restoration performance (PSNR) on a set of sequences, where σ is known. Then the restoration performance is maximized using the Nelder-Mead simplex direct search algorithm[18, 19] in a multivariate space, thus finding the optimum value of the triplet (γ_d, τ_{traj}, τ_{match}) for eight test video corrupted by i.i.d. white Gaussian noise having eight different value of σ, ranging from 5 to 70. Subsequently, we approximate the behavior of the three parameters as a function of σ using a quadratic polynomial for each variable in the domain (γ_d, τ_{traj}, τ_{match}) maximizing the total PSNR of the test sequences. The resulting fit is

$$\gamma_d(\sigma) = 0.0005 \cdot \sigma^2 - 0.0059 \cdot \sigma + 0.0400, \tag{22}$$

$$\tau_{\mathrm{traj}}(\sigma) = 0.0047 \cdot \sigma^2 + 0.0676 \cdot \sigma + 0.4564, \tag{23}$$

$$\tau_{\mathrm{match}}(\sigma) = 0.0171 \cdot \sigma^2 + 0.4520 \cdot \sigma + 47.9294. \tag{24}$$

The above functions are shown in Figure 3: experimentally they were found to be a good approximation of the optimum (γ_d, τ_{traj}, τ_{match}). Note that during the second stage such parameters can be considered constants independent of σ, because in the processed sequence $\hat{y}^{\mathrm{ht}}$ the noise is considerably lower than in the observation z; this is evident when looking at the second and third image of Figure 4. Moreover, since in this stage the

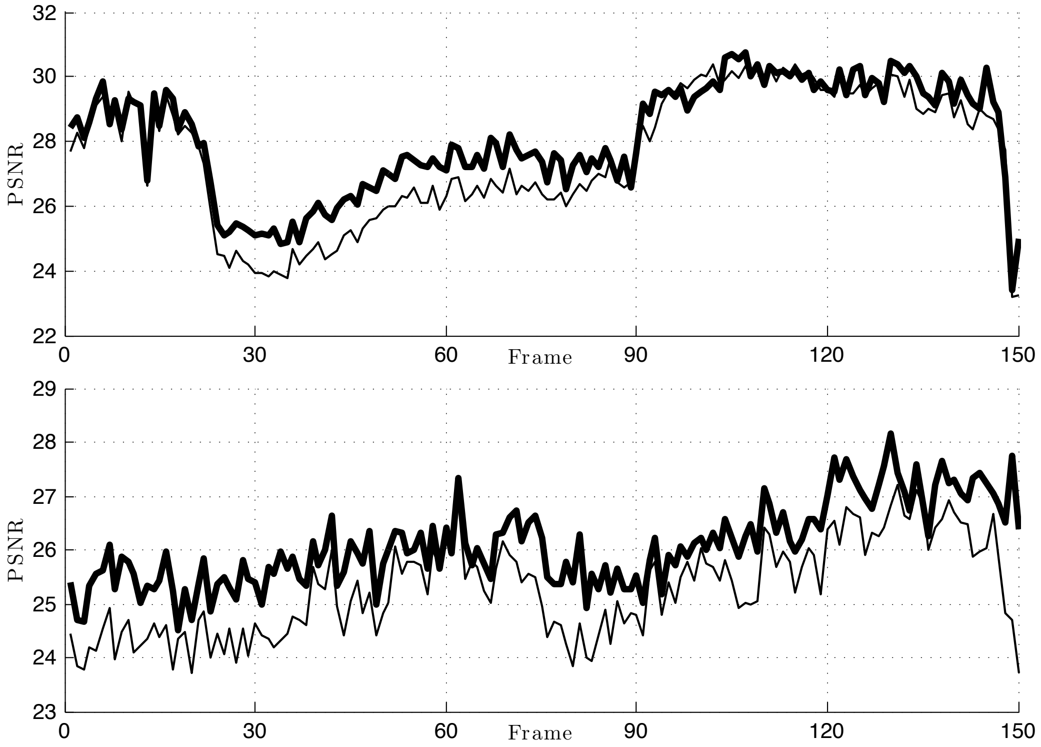


Figure 5. Frame-by-frame PSNR output of *Tennis* (top) and *Bus* (bottom) denoised by V-BM4D (thick line), and V-BM3D (thin line). The sequences are corrupted by i.i.d. white Gaussian noise with standard deviation $\sigma = 20$.

Table 2. Comparison between the PSNR (dB) outputs obtained from the proposed V-BM4D algorithm (top number in each cell), and the V-BM3D algorithm tuned with its default parameters[8] (bottom number in each cell). The test sequences are corrupted by i.i.d. Gaussian noise with zero mean and three different standard deviations σ.

σ	Video:	*Salesm.*	*Tennis*	*Fl. Gard.*	*Miss Am.*	*Coastg.*	*Foreman*	*Bus*	*Bicycle*
	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
10	V-BM4D	**37.30**	**35.22**	**32.81**	**40.09**	**35.54**	**36.94**	**34.26**	**37.66**
	V-BM3D	37.21	34.68	32.11	39.61	34.78	36.46	33.32	37.62
20	V-BM4D	33.79	**31.59**	**28.63**	**37.98**	**31.94**	**33.67**	**30.26**	34.10
	V-BM3D	**34.04**	31.20	28.24	37.85	31.71	33.30	29.57	**34.18**
40	V-BM4D	**30.35**	**28.49**	**24.60**	**35.47**	**28.54**	**30.52**	**26.72**	**30.10**
	V-BM3D	29.93	27.99	24.33	35.45	28.27	29.97	26.28	30.02

trajectories and the grouping are determined from the basic estimate $\hat{y}^{\text{ht}}$, there is no a straightforward relation with σ, the standard deviation of the noise corrupting the observation z.

The comparison against V-BM3D[8] is carried out using the set of parameters reported in Table 1. Table 2 compares the denoising performance in terms of PSNR of the two algorithms, applied to a set of standard video sequences corrupted by white Gaussian noise with increasing standard deviation $\sigma = \{10, 20, 40\}$, which is assumed known. Further details concerning the original sequences, such as the resolution and number of frames, are shown in the header of the table. As one can see, V-BM4D outperforms V-BM3D in nearly all the experiments, with PSNR improvement of up to 1 dB. It is particularly interesting to observe that V-BM4D handles effectively the sequences characterized by rapid motion and frequent scene changes, especially under

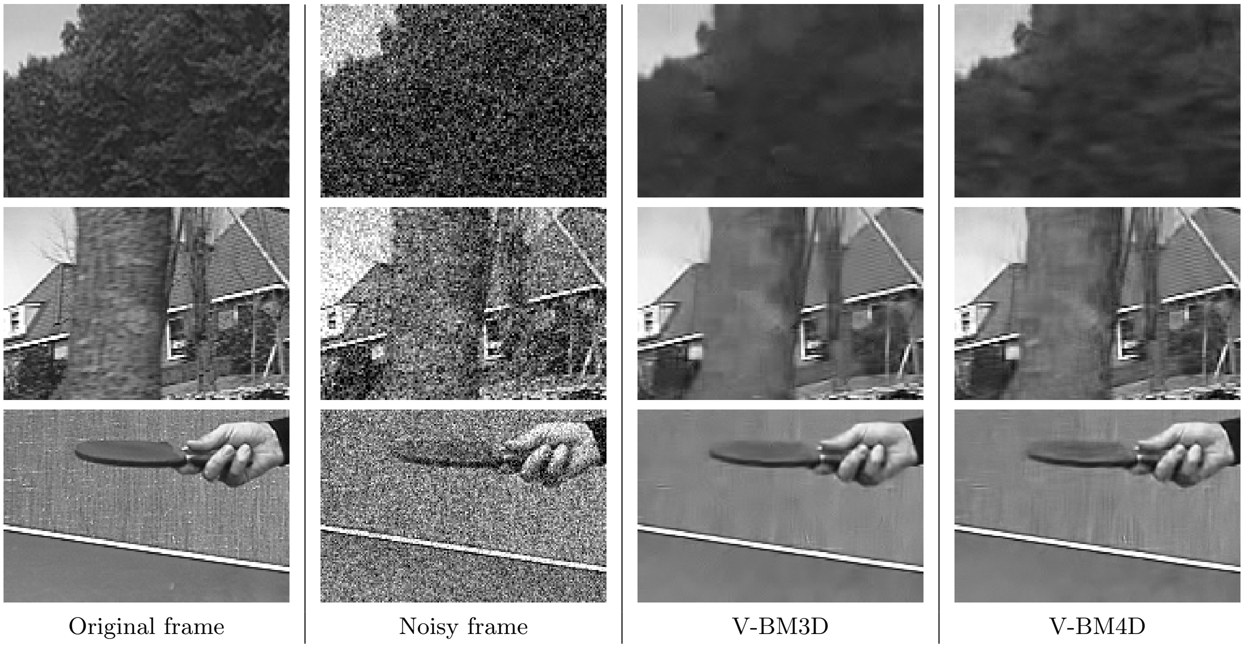


Figure 6. Visual comparison of the sequences, from top to bottom, *Bus*, *Flower Garden* and *Tennis* corrupted by white Gaussian noise with standard deviation $\sigma = 40$, denoised by the proposed algorithm V-BM4D and the V-BM3D algorithm.

heavy noise, as *Tennis*, *Flower Garden*, *Coastguard* and *Bus*. In particular, Figure 5 shows that as soon as the sequence presents a significant change in the scene, the denoising performance decreases significantly for the two algorithms, but, in these situations, V-BM4D requires much less frames to recover high PSNR values, as shown by the lower peaks at frame 30 and 90 of *Tennis* and around frame 75 of *Bus*.

Figure 6 offers a visual comparison of the performance of the two algorithms. As a subjective quality assessment, V-BM4D better preserves the textures, without introducing significant artifacts in the restored video: this is clearly visible in the tree leaves of the *Bus* sequence.

6. DISCUSSION AND CONCLUSIONS

Experiments show that V-BM4D outperforms V-BM3D in terms of measured performance (PSNR), and of visual appearance (Figure 6), thus achieving state-of-the-art results in video denoising. In particular, V-BM4D can restore much better than V-BM3D fine image details, even in sequences corrupted by heavy noise ($\sigma = 40$): this difference is clearly visible in the processed frames shown in Figure 6. Moreover, the comparison between V-BM3D and V-BM4D highlights that the temporal correlation is a key element in video denoising, and that it has to be adequately handled when designing nonlocal video restoration algorithms. We wish to remark that the computational complexity in V-BM4D is obviously higher than in V-BM3D, mainly because V-BM4D processes higher-dimensional arrays. Thus, V-BM4D can be a viable alternative to V-BM3D especially in applications where the highest restoration quality is paramount. Ongoing work addresses the parallelization of V-BM4D, leveraging GPU hardware.

REFERENCES

[1] Protter, M. and Elad, M., "Image sequence denoising via sparse and redundant representations," *IEEE Transactions on Image Processing* **18**(1), 27–35 (2009).

[2] Ghoniem, M., Chahir, Y., and Elmoataz, A., "Nonlocal video denoising, simplification and inpainting using discrete regularization on graphs," *Signal Processing* **90**(8), 2445–2455 (2010). Special Section on Processing and Analysis of High-Dimensional Masses of Image and Signal Data.

[3] Katkovnik, V., Foi, A., Egiazarian, K., and Astola, J., "From local kernel to nonlocal multiple-model image denoising," *International Journal of Computer Vision* **86**(1), 1–32 (2010).

[4] Buades, A., Coll, B., and Morel, J.-M., "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation* **4**(2), 490–530 (2005).

[5] Buades, A., Coll, B., and Morel, J.-M., "Nonlocal image and movie denoising," *Int. Journal of Computer Vision* **76**(2), 123–139 (2008).

[6] Li, X. and Zheng, Y., "Patch-based video processing: a variational bayesian approach," *IEEE Transactions on Circuits and Systems for Video Technology* **29**, 27–40 (January 2009).

[7] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K., "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. Image Process.* **16** (August 2007).

[8] Dabov, K., Foi, A., and Egiazarian, K., "Video denoising by sparse 3D transform-domain collaborative filtering," in [*Proc. 15th European Signal Processing Conference, EUSIPCO*], (September 2007).

[9] Boracchi, G. and Foi, A., "Multiframe raw-data denoising based on block-matching and 3-D filtering for low-light imaging and stabilization," in [*Proceedings of LNLA 2008, the International Workshop on Local and Non-Local Approximation in Image Processing, 22 - 24 August, 2008 Lausanne, Switzerland*], (2008).

[10] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K., "Joint image sharpening and denoising by 3D transform-domain collaborative filtering," in [*Proc. 2007 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2007*], (2007).

[11] Danielyan, A., Foi, A., Katkovnik, V., and Egiazarian, K., "Image and video super-resolution via spatially adaptive block-matching filtering," in [*Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA 2008, Lausanne, Switzerland*], (August 2008).

[12] Danielyan, A., Foi, A., Katkovnik, V., and Egiazarian, K., "Image upsampling via spatially adaptive block-matching filtering," in [*Proc. of 16th European Signal Processing Conference, EUSIPCO 2008*], (2008).

[13] Danielyan, A., Foi, A., Katkovnik, V., and Egiazarian, K., [*Spatially adaptive filtering as regularization in inverse imaging: compressive sensing, upsampling, and super-resolution, in Super-Resolution Imaging*], CRC Press / Taylor Francis (Sept. 2010).

[14] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K., "Image restoration by sparse 3D transform-domain collaborative filtering," in [*Proc. SPIE Electronic Imaging, San Jose (CA), USA*], **6812** (January 2008).

[15] Hang, H.-M., Chou, Y.-M., and Cheng, S.-C., "Motion estimation for video coding standards," *Journal of VLSI Signal Processing Systems* **17**(2/3), 113–136 (1997).

[16] Megret, R. and Dementhon, D., "A survey of spatio-temporal grouping techniques," tech. rep. (2002).

[17] Basharat, A., Zhai, Y., and Shah, M., "Content based video matching using spatiotemporal volumes," *Comput. Vis. Image Underst.* **110**(3), 360–377 (2008).

[18] Nelder, J. A. and Mead, R., "A simplex method for function minimization," *Computer Journal* **7**, 308–313 (1965).

[19] Lagarias, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E., "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM Journal of Optimization* **9**, 112–147 (1998).