# Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering

4 authors:

Kostadin Dabov
Apple Inc.
17 PUBLICATIONS   7,674 CITATIONS

SEE PROFILE

Alessandro Foi
Tampere University
106 PUBLICATIONS   12,430 CITATIONS

SEE PROFILE

Vladimir Katkovnik
Tampere University
337 PUBLICATIONS   14,304 CITATIONS

SEE PROFILE

Karen Egiazarian
Tampere University
680 PUBLICATIONS   21,406 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project   super-resolution View project

Project   Anomaly Detection in Images View project

# Image denoising by sparse 3D transform-domain collaborative filtering

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, *Senior Member, IEEE*

*Abstract*— We propose a novel image denoising strategy based on an enhanced sparse representation in transform domain. The enhancement of the sparsity is achieved by grouping similar 2D image fragments (e.g. blocks) into 3D data arrays which we call "groups". Collaborative filtering is a special procedure developed to deal with these 3D groups. We realize it using the three successive steps: 3D transformation of a group, shrinkage of the transform spectrum, and inverse 3D transformation. The result is a 3D estimate that consists of the jointly filtered grouped image blocks. By attenuating the noise, the collaborative filtering reveals even the finest details shared by grouped blocks and at the same time it preserves the essential unique features of each individual block. The filtered blocks are then returned to their original positions. Because these blocks are overlapping, for each pixel we obtain many different estimates which need to be combined. Aggregation is a particular averaging procedure which is exploited to take advantage of this redundancy. A significant improvement is obtained by a specially developed collaborative Wiener filtering. An algorithm based on this novel denoising strategy and its efficient implementation are presented in full detail; an extension to color-image denoising is also developed. The experimental results demonstrate that this computationally scalable algorithm achieves state-of-the-art denoising performance in terms of both peak signal-to-noise ratio and subjective visual quality.

*Index Terms*— image denoising, sparsity, adaptive grouping, block-matching, 3D transform shrinkage.

## I. INTRODUCTION

**P**LENTY of denoising methods exist, originating from various disciplines such as probability theory, statistics, partial differential equations, linear and nonlinear filtering, spectral and multiresolution analysis. All these methods rely on some explicit or implicit assumptions about the true (noise-free) signal in order to separate it properly from the random noise.

In particular, the transform-domain denoising methods typically assume that the true signal can be well approximated by a linear combination of few basis elements. That is, the signal is sparsely represented in the transform domain. Hence, by preserving the few high-magnitude transform coefficients that convey mostly the true-signal energy and discarding the rest which are mainly due to noise, the true signal can be effectively estimated. The sparsity of the representation depends on both the transform and the true-signal's properties.

The multiresolution transforms can achieve good sparsity for spatially localized details, such as edges and singularities.

Because such details are typically abundant in natural images and convey a significant portion of the information embedded therein, these transforms have found a significant application for image denoising. Recently, a number of advanced denoising methods based on multiresolution transforms have been developed, relying on elaborate statistical dependencies between coefficients of typically overcomplete (e.g. translation-invariant and multiply-oriented) transforms. Examples of such image denoising methods can be seen in [1], [2], [3], [4].

Not limited to the wavelet techniques, the overcomplete representations have traditionally played an important role in improving the restoration abilities of even the most basic transform-based methods. This is manifested by the sliding-window transform-domain image denoising methods [5], [6] where the basic idea is to apply shrinkage in local (windowed) transform domain. There, the overlap between successive windows accounts for the overcompleteness, while the transform itself is typically orthogonal, e.g. the 2D DCT.

However, the overcompleteness by itself is not enough to compensate for the ineffective shrinkage if the adopted transform cannot attain a sparse representation of certain image details. For example, the 2D DCT is not effective in representing sharp transitions and singularities, whereas wavelets would typically perform poorly for textures and smooth transitions. The great variety in natural images makes impossible for any fixed 2D transform to achieve good sparsity for all cases. Thus, the commonly used orthogonal transforms can achieve sparse representations only for particular image patterns.

The adaptive principal components of local image patches was proposed by Muresan and Parks [7] as a tool to overcome the mentioned drawbacks of standard orthogonal transforms. This approach produces good results for highly-structured image patterns. However, the computation of the correct PCA basis is essentially deteriorated by the presence of noise. With similar intentions, the K-SVD algorithm [8] by Elad and Aharon utilizes highly overcomplete dictionaries obtained via a preliminary training procedure. A shortcoming of these techniques is that both the PCA and learned dictionaries impose a very high computational burden.

Another approach [9] is to exploit a shape-adaptive transform on neighborhoods whose shapes are adaptive to salient image details and thus contain mostly homogeneous signal. The shape-adaptive transform can achieve a very sparse representation of the true signal in these adaptive neighborhoods.

Recently, an elaborate adaptive spatial estimation strategy, the *non-local means*, was introduced [10]. This approach is different from the transform domain ones. Its basic idea

is to build a pointwise estimate of the image where each pixel is obtained as a weighted average of pixels centered at regions that are similar to the region centered at the estimated pixel. The estimates are non-local as in principle the averages can be calculated over all pixels of the image. A significant extension of this approach is the exemplar-based estimator [11], which exploits pairwise hypothesis testing to define adaptive non-local estimation neighborhoods and achieves results competitive to the ones produced by the best transform-based techniques.

In this paper, we propose a novel image denoising strategy based on an enhanced sparse representation in transform-domain. The enhancement of the sparsity is achieved by *grouping* similar 2D fragments of the image into 3D data arrays which we call "groups". *Collaborative filtering* is a special procedure developed to deal with these 3D groups. It includes three successive steps: 3D transformation of a group, shrinkage of transform spectrum, and inverse 3D transformation. Thus, we obtain the 3D estimate of the group which consists of an array of jointly filtered 2D fragments. Due to the similarity between the grouped blocks, the transform can achieve a highly sparse representation of the true signal so that the noise can be well separated by shrinkage. In this way, the collaborative filtering reveals even the finest details shared by grouped fragments and at the same time it preserves the essential unique features of each individual fragment.

An image denoising algorithm based on this novel strategy is developed and described in detail. It generalizes and improves our preliminary algorithm introduced in [12]. A very efficient algorithm implementation offering effective complexity/performance trade-off is developed. Experimental results demonstrate that it achieves outstanding denoising performance in terms of both peak signal-to-noise ratio and subjective visual quality, superior to the current state-of-the-art. Extension to color-image denoising based on [13] is also presented.

The paper is organized as follows. We introduce the grouping and collaborative filtering concepts in Section II. The developed image denoising algorithm is described in Section III. An efficient and scalable realization of this algorithm can be found in Section IV and its extension to color-image denoising is given in Section V. Experimental results are presented in Section VI. Section VII gives an overall discussion of the developed approach and Section VIII contains relevant conclusions.

## II. GROUPING AND COLLABORATIVE FILTERING

We denominate *grouping* the concept of collecting similar $d$-dimensional fragments of a given signal into a $d+1$-dimensional data structure that we term "group". In the case of images for example, the signal fragments can be arbitrary 2D neighborhoods (e.g. image patches or blocks). There, a group is a 3D array formed by stacking together similar image neighborhoods. If the neighborhoods have the same shape and size, the formed 3D array is a generalized cylinder. The importance of grouping is to enable the use of a higher-dimensional filtering of each group, which exploits the potential similarity

(correlation, affinity, etc.) between grouped fragments in order to estimate the true signal in each of them. This approach we denominate *collaborative filtering*.

### A. Grouping

Grouping can be realized by various techniques; e.g., K-means clustering [14], self-organizing maps [15], fuzzy clustering [16], vector quantization [17], and others. There exist a vast literature on the topic; we refer the reader to [18] for a detailed and systematic overview of these approaches.

Similarity between signal fragments is typically computed as the inverse of some distance measure. Hence, a smaller distance implies higher similarity. Various distance measures can be employed, such as the $\ell^p$-norm of the difference between two signal fragments. Other examples are the weighted Euclidean distance ($p = 2$) used in the non-local means estimator [10], and also the normalized distance used in the exemplar-based estimator [11]. When processing complex or uncertain (e.g. noisy) data it might be necessary to first extract some features from the signal and then to measure the distance for these features only [18].

### B. Grouping by matching

Grouping techniques such as vector quantization or K-means clustering are essentially based on the idea of partitioning. It means that they build groups or clusters (classes) which are disjoint, in such a way that each fragment belongs to one and only one group. Constructing disjoint groups whose elements enjoy high mutual similarity typically requires recursive procedures and can be computationally demanding [18]. Furthermore, the partitioning causes unequal treatment of the different fragments because the ones that are close to the centroid of the group are better represented than those far from it. This happens always, even in the special case where all fragments of the signal are equidistantly distributed.

A much simpler and effective grouping of mutually similar signal fragments can be realized by *matching*, where in contrast to the above partitioning methods, the formed groups are not necessarily disjoint. Matching is a method for finding signal fragments similar to a given *reference* one. That is achieved by pairwise testing the similarity between the reference fragment and candidate fragments located at different spatial locations. The fragments whose distance (i.e. dissimilarity) from the reference one is smaller than a given threshold are considered mutually similar and are subsequently grouped. The similarity plays the role of the membership function for the considered group and the reference fragment can be considered as some sort of "centroid" for the group. Any signal fragment can be used as a reference one and thus a group can be constructed for it.

We remark that for most distance measures, establishing a bound on the distance between the reference fragment and all of the matched ones means that the distance between any two fragments in that group is also bounded. Roughly speaking, this bound is the diameter of the group. While for an arbitrary distance measure such a statement may not hold precisely,
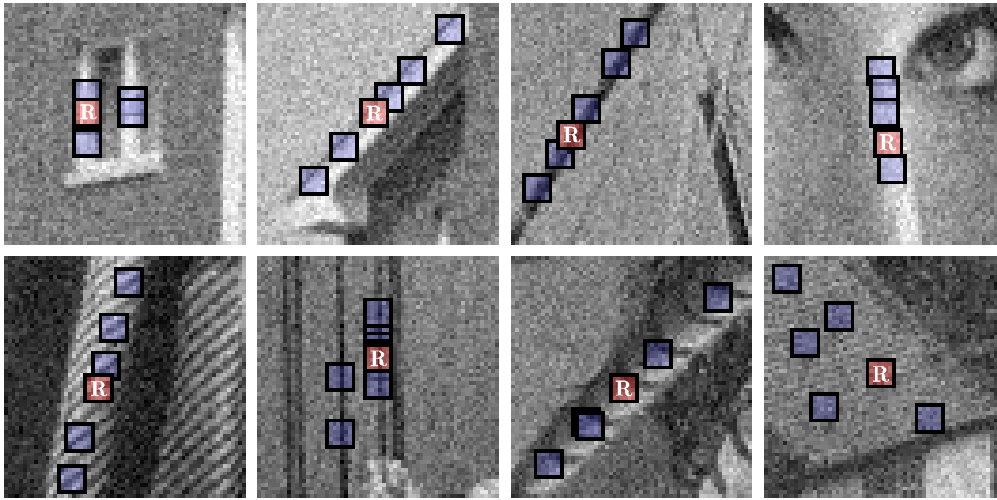
Fig. 1.   Illustration of grouping blocks from noisy natural images corrupted by white Gaussian noise with standard deviation 15 and zero mean. Each fragment shows a reference block marked with "R"and a few of the blocks matched to it.

for the case of metrics (e.g., $\ell^p$-norms) it is just a direct consequence of the triangle inequality.

Block-matching (BM) is a particular matching approach that has been extensively used for motion estimation in video compression (MPEG 1, 2, and 4, and H.26x). As a particular way of grouping, it is used to find similar blocks, which are then stacked together in a 3D array (i.e. a group). An illustrative example of grouping by block-matching for images is given in Figure 1, where we show a few reference blocks and the ones matched as similar to them.

### C. Collaborative filtering

Given a group of $n$ fragments, the collaborative filtering of the group produces $n$ estimates, one for each of the grouped fragments. In general, these estimates can be different. The term "collaborative" is taken literally, in the sense that each grouped fragment collaborates for the filtering of all others, and vice versa.

Let us consider an illustrative example of collaborative filtering for the estimation of the image in Figure 2 from an observation (not shown) corrupted by additive zero-mean independent noise. In particular, let us focus on the already grouped blocks shown in the same figure. These blocks exhibit perfect mutual similarity, which makes the elementwise averaging (i.e. averaging between pixels at the same relative positions) a suitable estimator. Hence, for each group, this collaborative averaging produces estimates of all grouped blocks. Because the corresponding noise-free blocks are assumed to be identical, the estimates are unbiased. Therefore, the final estimation error is due only to the residual variance which is inversely proportional to the number of blocks in the group. Regardless of how complex the signal fragments are, we can obtain very good estimates provided that the groups contain a large number of fragments.

However, perfectly identical blocks are unlikely in natural images. If non-identical fragments are allowed within the same group, the estimates obtained by elementwise averaging become biased. The bias error can account for the largest share
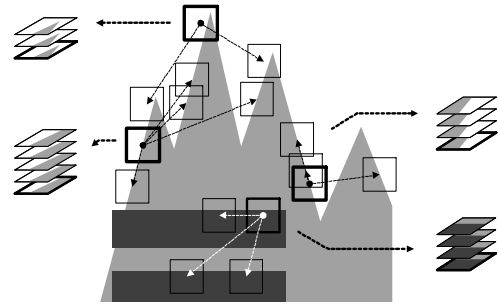


Fig. 2.   A simple example of grouping in an artificial image, where for each reference block (with thick borders) there exist perfectly similar ones.

of the overall final error in the estimates, unless one uses an estimator that allows for producing a different estimate of each grouped fragment. Therefore, a more effective collaborative filtering strategy than averaging should be employed.

### D. Collaborative filtering by shrinkage in transform domain

An effective collaborative filtering can be realized as shrinkage in transform domain. Assuming $d+1$-dimensional groups of similar signal fragments are already formed, the collaborative shrinkage comprises of the following steps.

- Apply a $d+1$-dimensional linear transform to the group.
- Shrink (e.g. by soft- and hard-thresholding or Wiener filtering) the transform coefficients to attenuate the noise.
- Invert the linear transform to produce estimates of all grouped fragments.

This collaborative transform-domain shrinkage can be particularly effective when applied to groups of natural image fragments, e.g. the ones in Figure 1. These groups are characterized by both:

○ intra-fragment correlation which appears between the pixels of each grouped fragment – a peculiarity of natural images;

○ inter-fragment correlation which appears between the corresponding pixels of different fragments – a result of

the similarity between grouped fragments.

The 3D transform can take advantage of both kinds of correlation and thus produce a sparse representation of the true signal in the group. This sparsity makes the shrinkage very effective in attenuating the noise while preserving the features of the signal.

Let us give a simple illustration of the benefit of this collaborative shrinkage by considering the grouped image blocks shown in Figure 1. Let us first consider the case when no collaborative filtering is performed but instead a 2D transform is applied separately to each individual block in a given group of $n$ fragments. Since these grouped blocks are very similar, for any of them we should get approximately the same number, say $\alpha$, of significant transform coefficients. It means that the whole group of $n$ fragments is represented by $n\alpha$ coefficients. In contrast, in the case of collaborative filtering, in addition to the 2D transform, we apply a 1D transform across the grouped blocks (equivalent to applying a separable 3D transform to the whole group). If this 1D transform has a DC-basis element, then because of the high similarity between the blocks, there are approximately[1] only $\alpha$ significant coefficients that represent the whole group instead of $n\alpha$. Hence, the grouping enhances the sparsity, which increases with the number of grouped blocks.

As Figure 1 demonstrates, a strong similarity between small image blocks at different spatial locations is indeed very common in natural images. It is a characteristic of blocks that belong to uniform areas, edges, textures, smooth intensity gradients, etc. Therefore, the existence of mutually similar blocks can be taken as a very realistic assumption when modeling natural images, which strongly motivates the use of grouping and collaborative filtering for an image denoising algorithm.

## III. ALGORITHM

In the proposed algorithm, the grouping is realized by block-matching and the collaborative filtering is accomplished by shrinkage in a 3D transform domain. The used image fragments are square blocks of fixed size. The general procedure carried out in the algorithm is as follows. The input noisy image is processed by successively extracting reference blocks from it and for each such block:

- find blocks that are similar to the reference one (block-matching) and stack them together to form a 3D array (group);
- perform collaborative filtering of the group and return the obtained 2D estimates of all grouped blocks to their original locations.

After processing all reference blocks, the obtained block estimates can overlap and thus there are multiple estimates for each pixel. We aggregate these estimates to form an estimate of the whole image.

This general procedure is implemented in two different forms to compose a two-step algorithm. This algorithm is illustrated in Figure 3 and proceeds as follows:

Step 1. *Basic estimate*.
   a) *Block-wise estimates*. For each block in the noisy image, do the following.
      i) *Grouping*. Find blocks that are similar to the currently processed one and then stack them together in a 3D array (group).
      ii) *Collaborative hard-thresholding*. Apply a 3D transform to the formed group, attenuate the noise by hard-thresholding of the transform coefficients, invert the 3D transform to produce estimates of all grouped blocks, and return the estimates of the blocks to their original positions.
   b) *Aggregation*. Compute the basic estimate of the true-image by weighted averaging all of the obtained block-wise estimates that are overlapping.

Step 2. *Final estimate*: using the basic estimate, perform improved grouping and collaborative Wiener filtering.
   a) *Block-wise estimates*. For each block, do the following.
      i) *Grouping*. Use BM within the basic estimate to find the locations of the blocks similar to the currently processed one. Using these locations, form two groups (3D arrays), one from the noisy image and one from the basic estimate.
      ii) *Collaborative Wiener filtering*. Apply a 3D transform on both groups. Perform Wiener filtering on the noisy one using the energy spectrum of the basic estimate as the true (pilot) energy spectrum. Produce estimates of all grouped blocks by applying the inverse 3D transform on the filtered coefficients and return the estimates of the blocks to their original positions.
   b) *Aggregation*. Compute a final estimate of the true-image by aggregating all of the obtained local estimates using a weighted average.

There are two significant motivations for the second step in the above algorithm:

- using the basic estimate instead of the noisy image allows to improve the grouping by block-matching;
- using the basic estimate as the pilot signal for the empirical Wiener filtering is much more effective and accurate than the simple hard-thresholding of the 3D spectrum of the noisy data.

*Observation model and notation*

We consider a noisy image $z : X \to \mathbb{R}$ of the form

$$z(x) = y(x) + \eta(x), \; x \in X,$$

where $x$ is a 2D spatial coordinate that belongs to the image domain $X \subset \mathbb{Z}^2$, $y$ is the true image, and $\eta$ is i.i.d. zero-mean Gaussian noise with variance $\sigma^2$, $\eta(\cdot) \sim \mathcal{N}(0, \sigma^2)$.

---

[1]This is just a qualitative statement because the actual number of significant coefficients depends on the normalization of the transforms and on the thresholds used for the 2D and 3D cases.
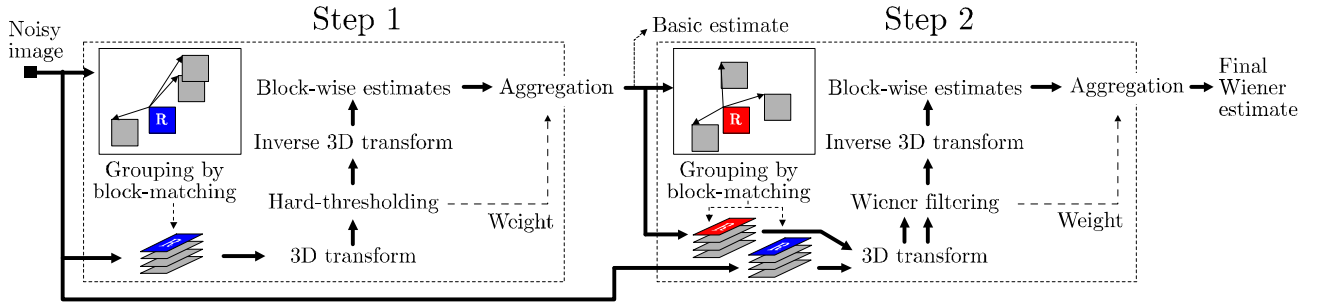
Fig. 3. Flowchart of the proposed image denoising algorithm. The operations surrounded by dashed lines are repeated for each processed block (marked with "R").

With $Z_x$ we denote a block of fixed size $N_1 \times N_1$ extracted from $z$, where $x$ is the coordinate of the top-left corner of the block. Alternatively, we say that $Z_x$ is located at $x$ in $z$. A group of collected 2D blocks is denoted by a bold-face capital letter with a subscript that is the set of its grouped blocks' coordinates, e.g., $\mathbf{Z}_S$ is a 3D array composed of blocks $Z_x$ located at $x \in S \subseteq X$. In order to distinguish between parameters used in the first and in the second step, we respectively use the superscripts "ht" (hard-thresholding) and "wie" (Wiener filtering). For example, $N_1^{\text{ht}}$ is the block size used in Step 1 and $N_1^{\text{wie}}$ is the block size used in Step 2. Analogously, we denote the basic estimate with $\widehat{y}^{\text{basic}}$ and the final estimate with $\widehat{y}^{\text{final}}$.

The following subsections present in detail the steps of the proposed denoising method.

### A. Steps 1a and 2a: Block-wise estimates

In this step, we process reference image blocks in a sliding-window manner. Here, "process" stands for performing grouping and estimating the true signal of all grouped blocks by:

- collaborative hard-thresholding in Step 1(a)ii,
- collaborative Wiener filtering in Step 2(a)ii.

The resultant estimates are denominated "block-wise estimates."

Because Steps 1a and 2a bear the same structure, we respectively present them in the following two sub-subsections. Therein, we fix the currently processed image block as $Z_{x_R}$ (located at the current coordinate $x_R \in X$) and denominate it "reference block."

*1) Steps 1(a)i and 1(a)ii: Grouping and collaborative hard-thresholding:* We realize grouping by block-matching within the noisy image $z$, as discussed in Section II-B. That is, only blocks whose distance (dissimilarity) with respect to the reference one is smaller than a fixed threshold are considered similar and grouped. In particular, we use the $\ell^2$-distance as a measure of dissimilarity.

Ideally, if the true-image $y$ would be available, the block-distance could be calculated as

$$d^{\text{ideal}}\left(Z_{x_R}, Z_x\right) = \frac{\left\|Y_{x_R} - Y_x\right\|_2^2}{\left(N_1^{\text{ht}}\right)^2}, \quad (1)$$

where $\|\cdot\|_2$ denotes the $\ell^2$-norm and the blocks $Y_{x_R}$ and $Y_x$ are respectively located at $x_R$ and $x \in X$ in $y$. However, only

the noisy image $z$ is available and the distance can only be calculated from the noisy blocks $Z_{x_R}$ and $Z_x$ as

$$d^{\text{noisy}}\left(Z_{x_R}, Z_x\right) = \frac{\left\|Z_{x_R} - Z_x\right\|_2^2}{\left(N_1^{\text{ht}}\right)^2}. \quad (2)$$

If the blocks $Z_{x_R}$ and $Z_x$ do not overlap, this distance is a non-central chi-squared random variable with mean

$$E\left\{d^{\text{noisy}}\left(Z_{x_R}, Z_x\right)\right\} = d^{\text{ideal}}\left(Z_{x_R}, Z_x\right) + 2\sigma^2$$

and variance

$$var\left\{d^{\text{noisy}}\left(Z_{x_R}, Z_x\right)\right\} = \frac{8\sigma^4}{\left(N_1^{\text{ht}}\right)^2} + \frac{8\sigma^2 d^{\text{ideal}}\left(Z_{x_R}, Z_x\right)}{\left(N_1^{\text{ht}}\right)^2}. \quad (3)$$

The variance grows asymptotically with $\mathcal{O}\left(\sigma^4\right)$. Thus, for relatively large $\sigma$ or small $N_1^{\text{ht}}$, the probability densities of the different $d^{\text{noisy}}\left(Z_{x_R}, Z_x\right)$ are likely to overlap heavily and this results in erroneous grouping[2]. That is, blocks with greater ideal distances than the threshold are matched as similar, whereas blocks with smaller such distances are left out.

To avoid the above problem, we propose to measure the block-distance using a coarse prefiltering. This prefiltering is realized by applying a normalized 2D linear transform on both blocks and then hard-thresholding the obtained coefficients, which results in

$$d\left(Z_{x_R}, Z_x\right) = \frac{\left\|\Upsilon'\left(\mathcal{T}_{2D}^{\text{ht}}\left(Z_{x_R}\right)\right) - \Upsilon'\left(\mathcal{T}_{2D}^{\text{ht}}\left(Z_x\right)\right)\right\|_2^2}{\left(N_1^{\text{ht}}\right)^2}, \quad (4)$$

where $\Upsilon'$ is the hard-thresholding operator with threshold $\lambda_{2D}\sigma$ and $\mathcal{T}_{2D}^{\text{ht}}$ denotes the normalized 2D linear transform[3].

Using the $d$-distance (4), the result of BM is a set that contains the coordinates of the blocks that are similar to $Z_{x_R}$,

$$S_{x_R}^{\text{ht}} = \left\{x \in X : d\left(Z_{x_R}, Z_x\right) \leq \tau_{\text{match}}^{\text{ht}}\right\}, \quad (5)$$

where the fixed $\tau_{\text{match}}^{\text{ht}}$ is the maximum $d$-distance for which two blocks are considered similar. The parameter $\tau_{\text{match}}^{\text{ht}}$ is selected from deterministic speculations about the acceptable value of the ideal difference, mainly ignoring the noisy components of the signal. Obviously $d\left(Z_{x_R}, Z_{x_R}\right) = 0$, which

---

[2]The effect of this is the sharp drop of the output-PSNR observed for two of the graphs in Figure 9 at about $\sigma = 40$.

[3]For simplicity, we do not invert the transform $\mathcal{T}_{2D}^{ht}$ and compute the distance directly from the spectral coefficients. When $\mathcal{T}_{2D}^{ht}$ is orthonormal, the distance coincides with the $\ell^2$-distance calculated between the denoised block-estimates in space domain.

implies that $\left| S_{x_R}^{\text{ht}} \right| \geq 1$, where $\left| S_{x_R}^{\text{ht}} \right|$ denotes the cardinality of $S_{x_R}^{\text{ht}}$. After obtaining $S_{x_R}^{\text{ht}}$, a group is formed by stacking the matched noisy blocks $Z_{x \in S_{x_R}^{\text{ht}}}$ to form a 3D array of size $N_1^{\text{ht}} \times N_1^{\text{ht}} \times \left| S_{x_R}^{\text{ht}} \right|$, which we denote $\mathbf{Z}_{S_{x_R}^{\text{ht}}}$. The matched blocks can in general overlap. We do not restrict the ordering, which is discussed in Section IV-B.

The collaborative filtering of $\mathbf{Z}_{S_{x_R}^{\text{ht}}}$ is realized by hard-thresholding in 3D transform domain. The adopted normalized 3D linear transform, denoted $\mathcal{T}_{\text{3D}}^{\text{ht}}$, is expected to take advantage of the two types of correlation, discussed in Section II-D, and attain good sparsity for the true signal group $\mathbf{Y}_{S_{x_R}^{\text{ht}}}$. This allows for effective noise attenuation by hard-thresholding, followed by inverse transform that yields a 3D array of block-wise estimates

$$\widehat{\mathbf{Y}}_{S_{x_R}^{\text{ht}}} = \mathcal{T}_{\text{3D}}^{\text{ht}^{-1}} \left( \Upsilon \left( \mathcal{T}_{\text{3D}}^{\text{ht}} \left( \mathbf{Z}_{S_{x_R}^{\text{ht}}} \right) \right) \right), \tag{6}$$

where $\Upsilon$ is a hard-threshold operator with threshold $\lambda_{\text{3D}}\sigma$. The array $\widehat{\mathbf{Y}}_{S_{x_R}^{\text{ht}}}$ comprises of $\left| S_{x_R}^{\text{ht}} \right|$ stacked block-wise estimates $\widehat{Y}_x^{\text{ht},x_R}$, $\forall x \in S_{x_R}^{\text{ht}}$. In $\widehat{Y}_x^{\text{ht},x_R}$, the subscript $x$ denotes the location of this block-estimate and the superscript $x_R$ indicates the reference block.

*2) Steps 2(a)i and 2(a)ii: Grouping and collaborative Wiener filtering:* Given the basic estimate $\widehat{y}^{\text{basic}}$ of the true image obtained in Step 1b, the denoising can be improved by performing grouping within this basic estimate and collaborative empirical Wiener filtering.

Because the noise in $\widehat{y}^{\text{basic}}$ is assumed to be significantly attenuated, we replace the thresholding-based $d$-distance (4) with the normalized squared $\ell^2$-distance computed within the basic estimate. This is a close approximation of the ideal distance (1). Hence, the coordinates of the matched blocks are the elements of the set

$$S_{x_R}^{\text{wie}} = \left\{ x \in X : \frac{\left\| \widehat{Y}_{x_R}^{\text{basic}} - \widehat{Y}_x^{\text{basic}} \right\|_2^2}{\left( N_1^{\text{wie}} \right)^2} < \tau_{\text{match}}^{\text{wie}} \right\}. \tag{7}$$

We use the set $S_{x_R}^{\text{wie}}$ in order to form two groups, one from the basic estimate and one from the noisy observation:

- $\widehat{\mathbf{Y}}_{S_{x_R}^{\text{wie}}}^{\text{basic}}$ by stacking together the basic estimate blocks $\widehat{Y}_{x \in S_{x_R}^{\text{wie}}}^{\text{basic}}$;
- $\mathbf{Z}_{S_{x_R}^{\text{wie}}}$ by stacking together the noisy blocks $Z_{x \in S_{x_R}^{\text{wie}}}$.

We define the empirical Wiener shrinkage coefficients from the energy of the 3D transform coefficients of the basic estimate group as

$$\mathbf{W}_{S_{x_R}^{\text{wie}}} = \frac{\left| \mathcal{T}_{\text{3D}}^{\text{wie}} \left( \widehat{\mathbf{Y}}_{S_{x_R}^{\text{wie}}}^{\text{basic}} \right) \right|^2}{\left| \mathcal{T}_{\text{3D}}^{\text{wie}} \left( \widehat{\mathbf{Y}}_{S_{x_R}^{\text{wie}}}^{\text{basic}} \right) \right|^2 + \sigma^2}. \tag{8}$$

Then the collaborative Wiener filtering of $\mathbf{Z}_{S_{x_R}^{\text{wie}}}$ is realized as the element-by-element multiplication of the 3D transform coefficients $\mathcal{T}_{\text{3D}}^{\text{wie}} \left( \mathbf{Z}_{S_{x_R}^{\text{wie}}} \right)$ of the noisy data with the Wiener shrinkage coefficients $\mathbf{W}_{S_{x_R}^{\text{wie}}}$. Subsequently, the inverse transform $\mathcal{T}_{\text{3D}}^{\text{wie}^{-1}}$ produces the group of estimates

$$\widehat{\mathbf{Y}}_{S_{x_R}^{\text{wie}}}^{wie} = \mathcal{T}_{\text{3D}}^{\text{wie}^{-1}} \left( \mathbf{W}_{S_{x_R}^{\text{wie}}} \mathcal{T}_{\text{3D}}^{\text{wie}} \left( \mathbf{Z}_{S_{x_R}^{\text{wie}}} \right) \right). \tag{9}$$

This group comprises of the block-wise estimates $\widehat{Y}_x^{wie,x_R}$ located at the matched locations $x \in S_{x_R}^{\text{wie}}$.

*B. Steps 1b and 2b: Global estimate by aggregation*

Each collection of block-wise estimates $\widehat{Y}_{x \in S_{x_R}^{\text{ht}}}^{\text{ht},x_R}$ and $\widehat{Y}_{x \in S_{x_R}^{\text{wie}}}^{\text{wie},x_R}$, $\forall x_R \in X$, obtained respectively in Steps 1a and 2a, is an overcomplete representation of the true-image because in general the block-wise estimates can overlap. In addition, more than one block-estimate can be located at exactly the same coordinate, e.g. $\widehat{Y}_{x_b}^{\text{ht},x_a}$ and $\widehat{Y}_{x_b}^{\text{ht},x_b}$ are both located at $x_b$ but obtained while processing the reference blocks at $x_a$ and $x_b$, respectively. One can expect substantially overcomplete representation of the signal in regions where there are plenty of overlapping block-wise estimates, i.e. where a block is matched (similar) to many others. Hence, the redundancy of the method depends on the grouping and therefore also on the particular image.

To compute the basic and the final estimates of the true-image in Steps 1b and 2b, respectively, we aggregate the corresponding block-wise estimates $\widehat{Y}_{x \in S_{x_R}^{\text{ht}}}^{\text{ht},x_R}$ and $\widehat{Y}_{x \in S_{x_R}^{\text{wie}}}^{\text{wie},x_R}$, $\forall x_R \in X$. This aggregation is performed by a weighted averaging at those pixel positions where there are overlapping block-wise estimates. The selection of weights is discussed in the following sub-section.

*1) Aggregation weights:* In general, the block-wise estimates are statistically correlated, biased, and have different variance for each pixel. However, it is quite demanding to take into consideration all these effects. Similarly to [6] and [9], we found that a satisfactory choice for aggregation weights would be ones that are inversely proportional to the total sample variance of the corresponding block-wise estimates. That is, noisier block-wise estimates should be awarded smaller weights. If the additive noise in the groups $\mathbf{Z}_{S_{x_R}^{\text{ht}}}$ and $\mathbf{Z}_{S_{x_R}^{\text{wie}}}$ is independent, the total sample variance in the corresponding groups of estimates (6) and (9) is respectively equal to $\sigma^2 N_{\text{har}}^{x_R}$ and $\sigma^2 \left\| \mathbf{W}_{S_{x_R}^{\text{wie}}} \right\|_2^2$, where $N_{\text{har}}^{x_R}$ is the number of retained (non-zero) coefficients after hard-thresholding and $\mathbf{W}_{S_{x_R}^{\text{wie}}}$ are the Wiener filter coefficients (8). Based on this, in Step 1b for each $x_R \in X$, we assign the weight

$$w_{x_R}^{\text{ht}} = \begin{cases} \frac{1}{\sigma^2 N_{\text{har}}^{x_R}}, & \text{if } N_{\text{har}}^{x_R} \geq 1 \\ 1, & \text{otherwise} \end{cases} \tag{10}$$

for the group of estimates $\widehat{Y}_{x \in S_{x_R}^{\text{ht}}}^{\text{ht},x_R}$. Similarly, in Step 2b for each $x_R \in X$, we assign the weight

$$w_{x_R}^{\text{wie}} = \sigma^{-2} \left\| \mathbf{W}_{S_{x_R}^{\text{wie}}} \right\|_2^{-2}, \tag{11}$$

for the group of estimates $\widehat{Y}_{x \in S_{x_R}^{\text{wie}}}^{\text{wie},x_R}$.

We remark that independence of the noise in a group is only achieved when the noisy blocks that build this group do not overlap each other. Therefore, on the one hand, the cost of ensuring independence would constitute a severe restriction for the BM, i.e. allowing matching only among non-overlapping blocks. On the other hand, if the possible overlaps are considered, the computation of the individual variance of

each transform coefficient in $\mathcal{T}_{3D}^{ht}\left(\mathbf{Z}_{S_{x_R}^{ht}}\right)$ or $\mathcal{T}_{3D}^{wie}\left(\mathbf{Z}_{S_{x_R}^{wie}}\right)$ becomes a prohibitive complication that requires considering the covariance terms in the corresponding transform coefficients. In our algorithm we use overlapping blocks but do not consider the covariances. Hence, the proposed weights (10) and (11) are only loosely inversely proportional to the corresponding total sample variances.

*2) Aggregation by weighted average:* The global basic estimate $\widehat{y}^{basic}$ is computed by a weighted average of the block-wise estimates $\widehat{Y}_{x \in S_{x_R}^{ht}}^{ht,x_R}$ obtained in Step 1a, using the weights $w_{x_R}^{ht}$ defined in (10), i.e.

$$\widehat{y}^{basic}\left(x\right) = \frac{\sum\limits_{x_R \in X} \sum\limits_{x_m \in S_{x_R}^{ht}} w_{x_R}^{ht} \widehat{Y}_{x_m}^{ht,x_R}\left(x\right)}{\sum\limits_{x_R \in X} \sum\limits_{x_m \in S_{x_R}^{ht}} w_{x_R}^{ht} \chi_{x_m}\left(x\right)}, \forall x \in X, \quad (12)$$

where $\chi_{x_m} : X \to \{0,1\}$ is the characteristic function of the square support of a block located at $x_m \in X$, and the block-wise estimates $\widehat{Y}_{x_m}^{ht,x_R}$ are zero-padded outside of their support.

The global final estimate $\widehat{y}^{final}$ is computed by (12), where $\widehat{y}^{basic}$, $\widehat{Y}_{x_m}^{ht,x_R}$, $S_{x_R}^{ht}$, and $w_{x_R}^{ht}$ are replaced respectively by $\widehat{y}^{final}$, $\widehat{Y}_{x_m}^{wie,x_R}$, $S_{x_R}^{wie}$, and $w_{x_R}^{wie}$.

## IV. FAST AND EFFICIENT REALIZATION

A straightforward implementation of the method presented in the previous section is highly computationally demanding. In order to realize a practical and efficient algorithm, we impose constraints and exploit certain expedients which we present in the following list.

Reduce the number of processed blocks.

- Rather than sliding by one pixel to every next reference block, use a step of $N_{step} \in \mathbb{N}$ pixels in both horizontal and vertical directions. Hence, the number of reference blocks is decreased from approximately $|X|$ to $\frac{|X|}{N_{step}^2}$.

Reduce the complexity of grouping.

- Restrict the maximum size of a group by setting an upper bound $N_2 \in \mathbb{N}$ on the number of grouped blocks; i.e. ensuring $|S_{x_R \in X}| \leq N_2$.
- Search for candidate matching blocks in a local neighborhood of restricted size $N_S \times N_S$ centered about the currently processed coordinate $x_R \in X$.
- To further speed-up the BM, we use *predictive search*, i.e. the search neighborhoods are non-rectangular and depend on the previously matched blocks. We form such a neighborhood as the union of $N_{PR} \times N_{PR}$ (where $N_{PR} \ll N_S$) ones centered at the previous matched coordinates correspondingly shifted by $N_{step}$ in the direction of processing the image, e.g. in horizontal direction for raster scan. For every $N_{FS}$-th processed block, we nevertheless perform an exhaustive-search BM in the larger $N_S \times N_S$ neighborhood. In particular, $N_{FS} = 1$ implies that only exhaustive-search in $N_S \times N_S$ is used.

Reduce the complexity of applying transforms.

- Restrict the transforms $\mathcal{T}_{3D}^{ht}$ and $\mathcal{T}_{3D}^{wie}$ to the class of separable transforms and use respectively $\mathcal{T}_{2D}^{ht}$ and $\mathcal{T}_{2D}^{wie}$

across the matched blocks and a 1D transform, $\mathcal{T}_{1D}$, along the third dimension of a group, along which the blocks are stacked.

- The spectra $\mathcal{T}_{2D}^{ht}\left(Z_x\right)$, $\mathcal{T}_{2D}^{wie}\left(Z_x\right)$, and $\mathcal{T}_{2D}^{wie}\left(\widehat{Y}_x^{basic}\right)$ are pre-computed for each block in a neighborhood $N_S \times N_S$ around the currently processed coordinate. Later, these are reused for subsequent reference blocks whose $N_S \times N_S$ neighborhoods overlap the current one. Thus, these transforms are computed exactly once for each processed coordinate; e.g. they are not re-computed each time in (4). In addition, in (6), (8), and (9), we compute the forward $\mathcal{T}_{3D}^{ht}$ and $\mathcal{T}_{3D}^{wie}$ transforms simply by applying $\mathcal{T}_{1D}$ across pre-computed $\mathcal{T}_{2D}^{ht}$- and $\mathcal{T}_{2D}^{wie}$-transformed blocks, respectively.

Realize efficiently the aggregation.

- First, in Steps 1(a)ii and 2(a)ii, the obtained block-wise estimates are weighted and accumulating in a buffer (with the size of the image). At the same time, the corresponding weights are accumulated at the same locations in another buffer. Then, in Steps 1b and 2b, the aggregation (12) is finally realized by a simple element-wise division between the two buffers.

Reduce the border effects.

- Use a $N_1 \times N_1$ Kaiser window (with parameter $\beta$) as part of the weights in (12) in order to reduce border effects which can appear when certain 2D transforms (e.g. the 2D DCT, the 2D DFT, or periodized wavelets) are used.

### A. Complexity

The time complexity of the algorithm is $\mathcal{O}\left(|X|\right)$ and thus depends linearly on the size of the input image, as all parameters are fixed.

Given the restrictions introduced in the previous subsection, without exploiting predictive-search BM, the number of operations per pixel is approximately

$$3\mathcal{C}_{\mathcal{T}_{2D}} + \frac{2\left(N_1^2 + N_2\right)N_S^2}{N_{step}^2} + \frac{3\left(N_2\mathcal{C}_{\mathcal{T}_{2D}} + N_1^2\mathcal{C}_{\mathcal{T}_{1D}}\right)}{N_{step}^2},$$

where for simplicity we omit the superscripts "ht" and "wie" from the parameters/operators, and where:

- the first addend is due to pre-computing $\mathcal{T}_{2D}$ for each sliding block (within a $N_S \times N_S$ neighborhood),
- the second is due to grouping by exhaustive-search BM in a $N_S \times N_S$ neighborhood, and
- the third addend is due to the transforms $\mathcal{T}_{3D}$ that is a separable composition of $\mathcal{T}_{2D}$ and $\mathcal{T}_{1D}$.

Above, $\mathcal{C}_{\mathcal{T}}$ denotes the number of arithmetic operations required for a transform $\mathcal{T}$; it depends on properties such as availability of fast algorithms, separability, etc. For example, the DFT can be computed efficiently by a fast Fourier transform algorithm and a dyadic wavelet decomposition can be realized efficiently using iterated filterbanks.

By exploiting fast separable transforms and the predictive-search BM, we can significantly reduce the complexity of the algorithm.

TABLE I
PARAMETER SETS FOR THE FAST AND NORMAL PROFILES.

| | | Fast Profile | Normal Profile | |
| --- | --- | --- | --- | --- |
| | | | $\sigma \leq 40$ | $\sigma > 40$ |
| Approx. exec. time for a 256×256 grayscale image on 1.5 GHz Celeron M | | 0.7 sec | 4.1 sec | 5.8 sec |
| Parameters for Step 1 (*ht*) | $\mathcal{T}_{2D}^{ht}$ | 2D-Bior1.5 | 2D-Bior1.5 | 2D-DCT |
| | $N_1^{ht}$ | 8 | 8 | 12 |
| | $N_2^{ht}$ | 16 | 16 | 16 |
| | $N_{step}^{ht}$ | 6 | 3 | 4 |
| | $N_S^{ht}$ | 25 | 39 | 39 |
| | $N_{FS}^{ht}$ | 6 | 1 | 1 |
| | $N_{PR}^{ht}$ | 3 | - | - |
| | $\beta^{ht}$ | 2.0 | 2.0 | 2.0 |
| | $\lambda_{2D}$ | 0 | 0 | 2.0 |
| | $\lambda_{3D}$ | 2.7 | 2.7 | 2.8 |
| | $\tau_{match}^{ht}$ | 2500 | 2500 | 5000 |
| Parameters for Step 2 (*wie*) | $\mathcal{T}_{2D}^{wie}$ | 2D-DCT | 2D-DCT | 2D-DCT |
| | $N_1^{wie}$ | 8 | 8 | 11 |
| | $N_2^{wie}$ | 16 | 32 | 32 |
| | $N_{step}^{wie}$ | 5 | 3 | 6 |
| | $N_S^{wie}$ | 25 | 39 | 39 |
| | $N_{FS}^{wie}$ | 5 | 1 | 1 |
| | $N_{PR}^{wie}$ | 2 | - | - |
| | $\tau_{match}^{wie}$ | 400 | 400 | 3500 |
| | $\beta^{wie}$ | 2.0 | 2.0 | 2.0 |
| Common | $\mathcal{T}_{1D}$ | 1D-Haar | 1D-Haar | 1D-Haar |

TABLE II
DEPENDENCY OF THE OUTPUT PSNR (DB) ON THE USED TRANSFORMS.
THE COLUMNS CORRESPONDING TO $\mathcal{T}_{2D}^{ht}$ CONTAIN PSNR RESULTS OF
THE BASIC ESTIMATE $\hat{y}^{basic}$ AND ALL OTHER COLUMNS CONTAIN RESULTS
OF THE FINAL ESTIMATE $\hat{y}^{final}$. THE NOISE IN THE OBSERVATIONS HAD
$\sigma = 25$.

| Transform | Boats | | | Lena | | |
| --- | --- | --- | --- | --- | --- | --- |
| | $\mathcal{T}_{2D}^{ht}$ | $\mathcal{T}_{2D}^{wie}$ | $\mathcal{T}_{1D}$ | $\mathcal{T}_{2D}^{ht}$ | $\mathcal{T}_{2D}^{wie}$ | $\mathcal{T}_{1D}$ |
| Haar | 29.31 | 29.84 | **29.91** | 31.24 | 31.93 | **32.08** |
| Db2 | 29.22 | 29.83 | 29.90 | 31.19 | 31.97 | 32.06 |
| Db4 | 29.34 | 29.88 | 29.89 | 31.31 | 32.01 | 32.06 |
| Db6 | 29.30 | 29.86 | 29.89 | 31.28 | 31.98 | 32.06 |
| Bior1.3 | 29.42 | 29.87 | 29.90 | 31.35 | 31.96 | 32.06 |
| Bior1.5 | **29.43** | 29.88 | 29.90 | 31.37 | 31.97 | 32.06 |
| WHT | 29.22 | 29.84 | 29.88 | 31.24 | 32.00 | 32.07 |
| DCT | 29.35 | **29.91** | 29.88 | **31.42** | **32.08** | 32.07 |
| DST | 29.33 | **29.91** | 29.79 | 31.36 | 31.97 | 31.92 |
| DC+rand | 29.07 | 29.75 | 29.88 | 31.06 | 31.88 | 32.06 |
| DC-only | - | - | 28.03 | - | - | 30.65 |

## B. Parameter selection

We studied the proposed algorithm using various transforms and parameters. As a results, we propose sets of parameters that are categorized in two profiles, "Normal" and "Fast", presented in Table I. The main characteristics of these profiles are as follows.

- Normal Profile. This profile offers a reasonable compromise between computational complexity and denoising performance. It is divided in two cases depending on the level of noise:

   $\sigma \leq 40$   the noise is not too severe to affect the correctness of the grouping, hence the thresholding in the $d$-distance (4) is disabled by setting $\lambda_{2D} = 0$ and relatively small block sizes are used, $N_1^{ht}, N_1^{wie} = 8$.

   $\sigma > 40$   corresponds to high level of noise, hence $\lambda_{2D} = 2$ is used to improve the correctness of the grouping and larger block sizes are used, $N_1^{ht} = 12$ and $N_1^{wie} = 11$.

- Fast Profile. Provides lower computational complexity at the cost of decreased denoising performance. It exploits the proposed fast predictive-search BM (unlike the Normal Profile, which uses only the exhaustive-search BM).

The benefit of using thresholding ($\lambda_{2D} = 2$) for the $d$-distance and larger block sizes when $\sigma > 40$ is illustrated in Figure 9 and discussed in Section VI.

To show how the denoising performance depends on the choice of the transforms $\mathcal{T}_{2D}^{ht}$, $\mathcal{T}_{2D}^{wie}$, and $\mathcal{T}_{1D}$, we present some experimental results in Table II. As already stated, the 3D transforms $\mathcal{T}_{3D}^{ht}$ and $\mathcal{T}_{3D}^{wie}$ used in Steps 1 and 2 of our method are formed by a separable composition of $\mathcal{T}_{2D}^{ht}$ and $\mathcal{T}_{2D}^{wie}$,

respectively, with $\mathcal{T}_{1D}$. Furthermore, both $\mathcal{T}_{2D}^{ht}$ and $\mathcal{T}_{2D}^{wie}$ are separable compositions of 1D transforms such as the ones specified in the table. The following normalized transforms were used in our experiment:

- DST, DCT: the discrete sine and cosine transforms,
- WHT: the Walsh-Hadamard transform, and
- a few full dyadic wavelet decompositions using:
  - Db$p$: the Daubechies wavelet with $p$ vanishing moments, where $p = 1, 2, 4, 6$; when $p = 1$, it coincides with the Haar wavelet,
  - Bior1.$N_r$: a bi-orthogonal spline wavelet, where the vanishing moments of the decomposing and the reconstructing wavelet functions are 1 and $N_r$, respectively,
- DC+rand: an orthonormal transform that has a DC basis element and the rest of its basis elements have random nature, i.e. obtained by orthonormalization of realizations of a white Gaussian process.

In addition, only for $\mathcal{T}_{1D}$, we experimented with elementwise averaging, i.e. preserving only the DC in the third dimension (and discarding all other transform coefficients), hence its name "DC-only". For this case, all grouped blocks are estimated by elementwise averaging, exactly as in the illustrative example of Section II-C.

In Table II, we present results corresponding to various $\mathcal{T}_{2D}^{ht}$, $\mathcal{T}_{2D}^{wie}$, and $\mathcal{T}_{1D}$. There, the Normal Profile parameters were used in all cases, where only the transform corresponding to a particular table column was changed. Boldface result correspond to the best performing transform. We observe that the choice of $\mathcal{T}_{2D}^{ht}$ and $\mathcal{T}_{2D}^{wie}$ does not have a significant impact on the denoising performance. Even the "DC+rand" transform, whose basis elements except for the DC are random, shows only a modest PSNR decrease in the range $0.1 - 0.4$ dB. This can be explained by the fact that the collaborative filtering depends mainly on $\mathcal{T}_{1D}$ for exploiting the inter-fragment correlation among grouped blocks. The estimation ability does not significantly depend on the energy compaction capabilities of $\mathcal{T}_{2D}^{ht}$ and $\mathcal{T}_{2D}^{wie}$. In this sense, the inter-fragment correlation appears as a much more important feature than the intra-

fragment correlation.

Let us now focus on the results corresponding to the various $\mathcal{T}_{1D}$ transforms in Table II. One can distinguish the moderately worse performance of the DST as compared with not only the other standard transforms but also with the "DC+rand". We argue that the reason for this is the lack of DC basis element in the DST — in contrast with all other transforms, which have this element. Why is the DC of $\mathcal{T}_{1D}$ important? Roughly speaking, this is so because the DC basis element captures the similarity between elements along the 3rd dimension of a group. Since the grouped blocks are similar, so are their corresponding 2D spectra and the DC terms reflect this similarity.

However, as it has been discussed in Section II, the existence of perfectly matching blocks is unlikely. In order to avoid trivial groups containing only the reference block, a strictly positive threshold is used in Equations (5) and (7). Additionally, as follows from Equation (3), the accuracy of the block-distance is affected by the noise. In practice this means that within a group there can be blocks for which the underlying true signal $Y_x$ is much farther from $Y_{x_R}$ than $\tau^{ht}_{match}$. Therefore, the sole DC element is not able to capture the potential differences between grouped blocks. This is confirmed by the poor results of the "DC-only" for $\mathcal{T}_{1D}$. The availability of additional basis elements in any of the other transforms, even the random ones in "DC+rand", results in big performance improvement over the "DC-only".

We experimentally found that the ordering of blocks in the group does not have a significant effect on the estimation ability of the algorithm. This is confirmed by the results of the "DC+rand" for $\mathcal{T}_{1D}$ which achieves the same results as any of the other (structured, non-random) orthogonal transforms. For this transform, the ordering is *irrelevant* for the DC and is relevant only for the other basis elements which however are generated randomly. Hence, we may conclude that the ordering of the blocks in the groups does not influence the final results. Given this and because in our implementation the BM already produces a collection of blocks ordered by their block-distance, we resort to using exactly this ordering. Naturally, first in a group is always the reference block as the distance to itself is trivially equal to zero.

Note that even though a group is constructed based on the similarity with respect to a given reference block, this does not imply that this block is better represented by the group than any of the others. For example, it can happen that all the matched blocks (except the reference block) are quite dissimilar from the reference one but tightly similar to each other. Such a group could be termed as "unbalanced."

We choose the Haar full dyadic decomposition for $\mathcal{T}_{1D}$ because it can be efficiently implemented with iterated filterbanks using 2-tap analysis/synthesis filters. To apply such an orthonormal full dyadic decomposition, the transform size must be a power of 2. We enforced this requirement by restricting the number of elements of both $S^{ht}_x$ (5) and $S^{wie}_x$ (7) to be the largest power of 2 smaller than or equal to the original number of elements in $S^{ht}_x$ and $S^{wie}_x$, respectively.

## V. EXTENSION TO COLOR-IMAGE DENOISING

We consider a natural *RGB* image with additive i.i.d. zero-mean Gaussian noise in each of its channels. Let a luminance-chrominance transformation be applied on such a noisy image, where the luminance channel is denoted with $Y$ and the chrominance channels are denoted with $U$ and $V$. Prominent examples of such transformations are the *YCbCr* and the opponent color transformations, whose transform matrices are respectively

$$A_{YCbCr} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ -0.17 & -0.33 & 0.50 \\ 0.50 & -0.42 & -0.08 \end{bmatrix}, \quad A_{opp} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{\sqrt{6}} & 0 & \frac{-1}{\sqrt{6}} \\ \frac{1}{3\sqrt{2}} & \frac{-\sqrt{2}}{3} & \frac{1}{3\sqrt{2}} \end{bmatrix}. \quad (13)$$

Due to properties of the underlying natural color image, such as high correlation between its $R$, $G$, and $B$ channels, the following observations can be made:

- $Y$ has higher SNR than $U$ and $V$ (decorrelation of the $R$, $G$, and $B$ channels);
- $Y$ contains most of the valuable information (edges, shades, objects, texture patterns, etc.);
- $U$ and $V$ contain mostly low-frequency information (very often these channels come from undersampled data);
- iso-luminant regions with variation only in $U$ and $V$ are unlikely.

A straightforward extension of the developed grayscale denoising method for color-image denoising would be to apply it separately on each of the $Y$, $U$, and $V$ channels. This naive approach however would suffer from the lower SNR in the chrominances since the grouping is sensitive to the level of noise. Because a proper grouping is essential for the effectiveness of our method, we propose to perform the grouping only once for the luminance $Y$ and reuse exactly the same grouping when applying collaborative filtering on the chrominances $U$ and $V$. That is, the sets of grouped blocks' coordinates from Eq. (5) and (7) are found for $Y$, respectively in Steps 1(a)i and 2(a)i, and reused for both $U$ and $V$; using these sets, the collaborative filtering (Steps 1(a)ii and 2(a)ii) and the aggregation (Steps 1b and 2b) are performed separately on each of the three channels. The *grouping constraint* on the chrominances is based on the assumption that if the luminances of two blocks are mutually similar, then their chrominances are also mutually similar. Furthermore, given that grouping by block-matching takes approximately half of the execution time of the BM3D, the grouping constraint enables a computational reduction of approximately one third as compared to applying the grayscale BM3D separately on the three channels.

## VI. RESULTS

In this section we present and discuss the experimental results obtained by the developed algorithms; the grayscale version is denominated *Block-Matching and 3D filtering* (BM3D) and the color version is accordingly abbreviated C-BM3D. For all experiments we used the Matlab codes available at `http://www.cs.tut.fi/~foi/GCF-BM3D`. At this website we also provide further results and the original and

denoised test images used in our experiments. Unless specified otherwise, we use the parameters of the "Normal Profile" from Table I for both the BM3D and the C-BM3D.

## A. Grayscale-image denoising

The output PSNR results of the BM3D algorithm for a standard set of grayscale images are given in Table III. The PSNR of an estimate $\widehat{y}$ of a true image $y$, is computed according to the standard formula

$$\text{PSNR}\left(\widehat{y}\right) = 10\log_{10}\left(\frac{255^2}{|X|^{-1}\sum_{x \in X}\left(y\left(x\right) - \widehat{y}\left(x\right)\right)^2}\right).$$

In Figure 4, we compare the output PSNR results of the proposed BM3D with those of the state-of-the-art techniques BLS-GSM [3], FSP+TUP BLS-GSM [4], exemplar-based [11], K-SVD [8], Pointwise SA-DCT [9]; for the K-SVD method [8] we report its best results, which are those obtained with an adaptive dictionary trained on the noisy image. It can be seen from the figure that the proposed BM3D demonstrates the best performance and uniformly outperforms all of the other techniques. In particular, a significant improvement is observed for *House* and *Barbara* since these images contain structured objects (edges in *House* and textures in *Barbara*) which enable a very effective grouping and collaborative filtering.

In Figure 5, we show a noisy ($\sigma = 25$) *House* image and the corresponding BM3D estimate. In this test image, similarity among neighboring blocks is easy to perceive in the uniform regions and along the regular-shaped structures, some of which are illustrated in Figure 1. Hence, such details are well-preserved in the estimate.

The denoising performance of the BM3D algorithm is further illustrated in Figure 6, where we show fragments of a few noisy ($\sigma = 25$) test images and fragments of the corresponding denoised ones. The denoised images show good preservation of:

- uniform areas and smooth intensity transitions (cheeks of *Lena*, and the backgrounds of the other images),
- textures and repeating patterns (the scarf in *Barbara*), and
- sharp edges and singularities (borders of objects in *Cameraman* and *Boats*).

A denoising example for an extreme level of noise such as $\sigma = 100$ is shown in Figure 7. Given that the original image is almost completely buried into noise, the produced estimate shows reasonable detail preservation. In particular, repeated patterns such as the stripes on the clothes are faithfully reconstructed.

Regarding the subjective visual quality, we find that various image details are well preserved and at the same time very few artifacts are introduced; one can observe this in Figures 6, 7, and 8. The state-of-the-art subjective visual quality of our algorithm is confirmed by the result of the psycho-visual experiment carried out by Vansteenkiste et al. [19]. There, 35 evaluators classified the preliminary version [12] of the BM3D algorithm as the best among 8 evaluated state-of-the-art techniques. The criteria in this evaluation were *perceived noisiness*, *perceived blurriness*, and *overall visual quality*. Furthermore, we consider the subjective visual quality of the

current BM3D algorithm to be significantly better (in terms of detail preservation) than that of its preliminary version evaluated in [19]. In Figure 8 we show images denoised by the current and by the preliminary versions of the BM3D algorithm. A close inspection reveals that the images denoised by the current BM3D have both fewer ringing artifacts and better preservation of details.

We show the PSNR performance of the Fast and Normal BM3D Profiles in Figure 9. The two cases of the Normal Profile from Table I are considered separately for $\sigma \in [10, 75]$ in order to show the sharp PSNR drop of the "$\sigma \leq 40$" graph at about $\sigma = 40$ due to erroneous grouping. On the other hand, for the "$\sigma > 40$" graph, where the thresholding-based $d$-distance (4) is used with a relatively large block-size $N_1$, one can observe that there is no sharp PSNR drop. It is noteworthy that for up to moderate levels of noise, such as $\sigma < 35$, the PSNR difference between the Fast and the Normal Profiles is in the range $0.05 - 0.2$ dB. This can be an acceptable price for the 6-fold reduction of the execution time shown in Table I; more precisely, the approximate execution time (for denoising a 256×256 image calculated on a 1.5 GHz Celeron M) decreases from 4.1 seconds for the Normal Profile to 0.7 seconds for the Fast Profile. The BM3D algorithm allows for further complexity/performance trade-off by varying $N_{\text{step}}$. As a rough comparison, the execution times (for denoising a 256×256 image on a 1.5GHz Celeron M) of the other methods considered in Figure 4 were: 22.1 seconds for the BLS-GSM, 6.2 seconds for the SA-DCT filter, $9 - 30$ minutes (depending on $\sigma$) for training the adaptive K-SVD on an input noisy image and $25 - 120$ seconds to perform the filtering using the found dictionary. The execution time of the exemplar-based method was reported in [11] to be about 1 minute when measured on a 2 GHz Pentium IV. The execution time of the FSP+TUP BLS-GSM was not reported; however, it is a two-step BLS-GSM extension that should not be faster than the BLS-GSM.

## B. Color-image denoising

We performed experiments with the C-BM3D using the opponent color space transformation (13) and the Normal Profile algorithm parameters. In all experiments, we considered noisy images with i.i.d. zero-mean Gaussian noise of variance $\sigma^2$ in each of their $R$, $G$, and $B$ channels. The PSNR for *RGB* images is computed using the standard formula

$$10\log_{10}\left(\frac{255^2}{(3\,|X|)^{-1}\sum_{c=R,G,B}\sum_{x \in X}\left(y_c\left(x\right) - \widehat{y}_c\left(x\right)\right)^2}\right),$$

where the subscript $c \in \{R, G, B\}$ denotes the color channel. Table IV presents the output-PSNR results of the proposed C-BM3D algorithm for a few standard test images. A comparison with the two recent state-of-the-art methods [9], [20] is given in Table V. One can see that the proposed algorithm outperforms them for the three test images considered there. The visual quality can be inspected from Figure 10 where a noisy (with $\sigma = 50$) color *Lena* and the C-BM3D estimate are shown. One can observe the faithfully preserved details on the hat, the sharp edges, and the smooth regions. The approximate

TABLE III

GRAYSCALE-IMAGE DENOISING: OUTPUT PSNR (DB) OF THE PROPOSED BM3D ALGORITHM.

| $\sigma$ / PSNR | C.man $256^2$ | House $256^2$ | Peppers $256^2$ | Montage $256^2$ | Lena $512^2$ | Barbara $512^2$ | Boats $512^2$ | F.print $512^2$ | Man $512^2$ | Couple $512^2$ | Hill $512^2$ | Lake $512^2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 / 42.11 | 43.96 | 44.63 | 43.48 | 46.47 | 43.59 | 43.66 | 43.18 | 42.90 | 43.61 | 43.17 | 43.04 | 43.02 |
| 5 / 34.16 | 38.29 | 39.83 | 38.12 | 41.14 | 38.72 | 38.31 | 37.28 | 36.51 | 37.82 | 37.52 | 37.14 | 36.58 |
| 10 / 28.14 | 34.18 | 36.71 | 34.68 | 37.35 | 35.93 | 34.98 | 33.92 | 32.46 | 33.98 | 34.04 | 33.62 | 32.85 |
| 15 / 24.61 | 31.91 | 34.94 | 32.70 | 35.15 | 34.27 | 33.11 | 32.14 | 30.28 | 31.93 | 32.11 | 31.86 | 31.08 |
| 20 / 22.11 | 30.48 | 33.77 | 31.29 | 33.61 | 33.05 | 31.78 | 30.88 | 28.81 | 30.59 | 30.76 | 30.72 | 29.87 |
| 25 / 20.18 | 29.45 | 32.86 | 30.16 | 32.37 | 32.08 | 30.72 | 29.91 | 27.70 | 29.62 | 29.72 | 29.85 | 28.94 |
| 30 / 18.59 | 28.64 | 32.09 | 29.28 | 31.37 | 31.26 | 29.81 | 29.12 | 26.83 | 28.86 | 28.87 | 29.16 | 28.18 |
| 35 / 17.25 | 27.93 | 31.38 | 28.52 | 30.46 | 30.56 | 28.98 | 28.43 | 26.09 | 28.22 | 28.15 | 28.56 | 27.50 |
| 50 / 14.16 | 25.84 | 29.37 | 26.41 | 27.35 | 28.86 | 27.17 | 26.64 | 24.36 | 26.59 | 26.38 | 27.08 | 25.78 |
| 75 / 10.63 | 24.05 | 27.20 | 24.48 | 25.04 | 27.02 | 25.10 | 24.96 | 22.68 | 25.10 | 24.63 | 25.58 | 24.11 |
| 100 / 8.14 | 22.81 | 25.50 | 22.91 | 23.38 | 25.57 | 23.49 | 23.74 | 21.33 | 23.97 | 23.37 | 24.45 | 22.91 |
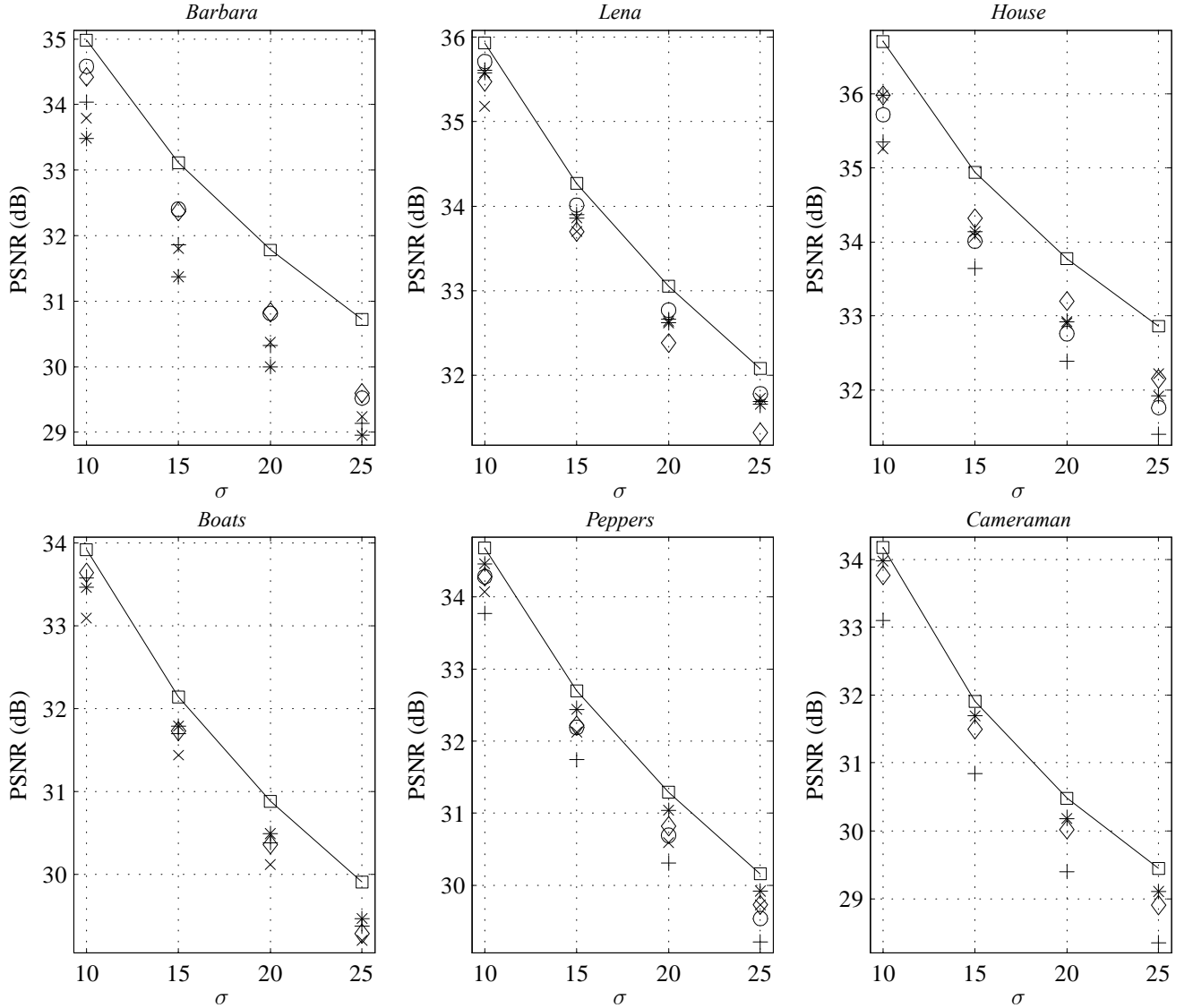


Fig. 4.   Grayscale-image denoising: output PSNR as a function of $\sigma$ for the following methods: '□': proposed BM3D, 'O': FSP+TUP BLS-GSM [4], '+': BLS-GSM [3], '×': exemplar-based [11], '◇': K-SVD [8], '✳': Pointwise SA-DCT [9]. (Note that the result of [4] for *Boats* and the results of [4], [11] for *Cameraman* are missing since they were neither reported in the corresponding articles, nor were implementations of these methods publicly available.)

Fig. 5.   Noisy ($\sigma = 25$) grayscale *House* image and the BM3D estimate (PSNR 32.86 dB).
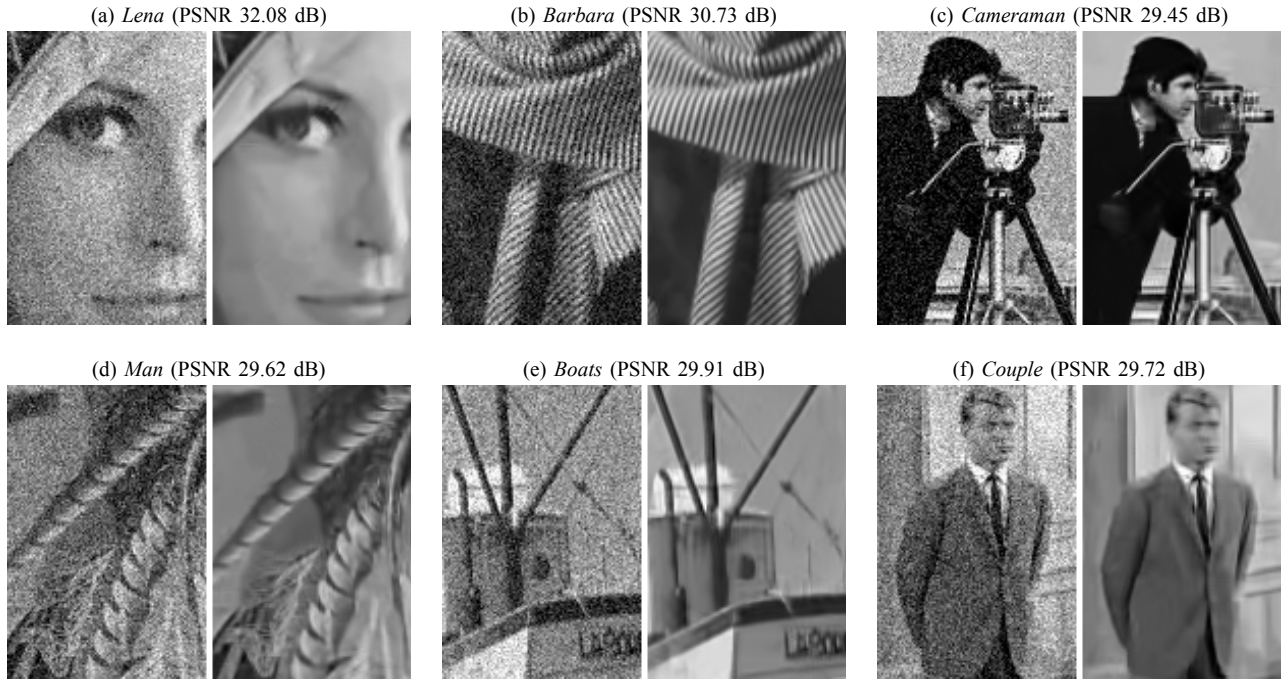


(a) *Lena* (PSNR 32.08 dB)  (b) *Barbara* (PSNR 30.73 dB)  (c) *Cameraman* (PSNR 29.45 dB)

(d) *Man* (PSNR 29.62 dB)  (e) *Boats* (PSNR 29.91 dB)  (f) *Couple* (PSNR 29.72 dB)

Fig. 6.   Fragments of noisy ($\sigma = 25$, PSNR 20.18 dB) grayscale images and the corresponding BM3D estimates.

execution time of the C-BM3D for a $256 \times 256$ RGB image was 7.6 seconds on a 1.5 GHz Celeron M.

## VII. DISCUSSION

The approach presented in this paper is an evolution of our work on local approximation techniques. It started from the classical local polynomial approximation with a simple symmetric neighborhood. The adaptive pointwise varying size of this neighborhood was a first step to practically efficient algorithms. A next step was devoted to anisotropic estimation based on adaptive starshaped neighborhoods allowing non-symmetric estimation areas. The non-symmetry of these estimates is a key-point in designing estimators relevant to natural

TABLE IV
COLOR-IMAGE DENOISING: OUTPUT PSNR OF THE PROPOSED C-BM3D ALGORITHM.

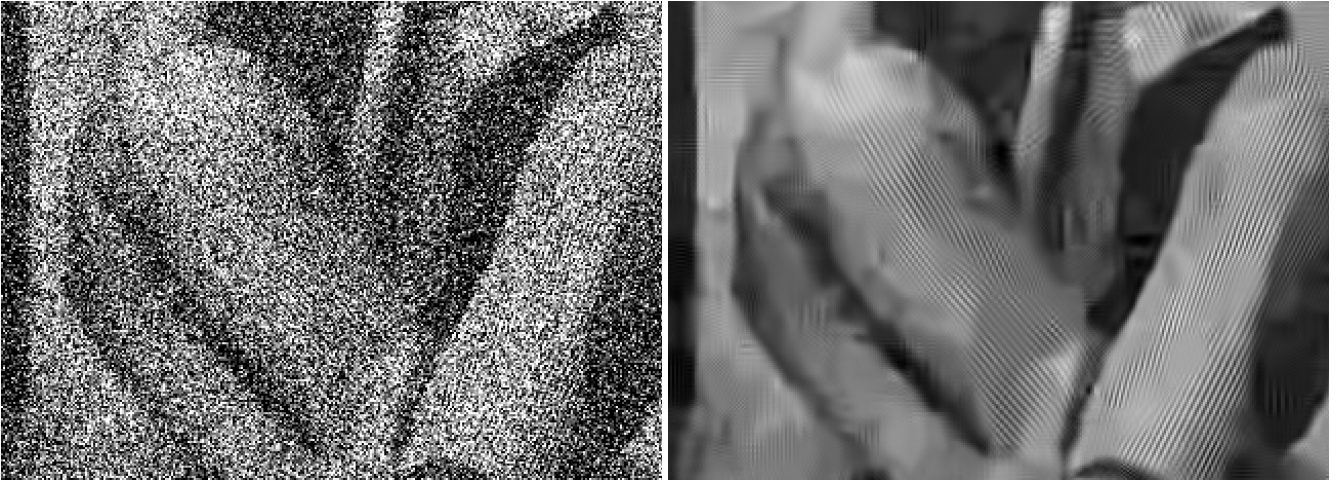| $\sigma$ / PSNR | Lena | Peppers | Baboon | F16 | House |
|---|---|---|---|---|---|
| 5 / 34.15 | 37.82 | 36.82 | 35.25 | 39.68 | 38.97 |
| 10 / 28.13 | 35.22 | 33.78 | 30.64 | 36.69 | 36.23 |
| 15 / 24.61 | 33.94 | 32.60 | 28.39 | 35.00 | 34.85 |
| 20 / 22.11 | 33.02 | 31.83 | 26.97 | 33.77 | 33.84 |
| 25 / 20.17 | 32.27 | 31.20 | 25.95 | 32.78 | 33.03 |
| 30 / 18.59 | 31.59 | 30.62 | 25.14 | 31.94 | 32.33 |
| 35 / 17.25 | 30.91 | 30.00 | 24.46 | 31.13 | 31.58 |
| 50 / 14.15 | 29.72 | 28.68 | 23.14 | 29.41 | 30.22 |
| 75 / 10.63 | 28.19 | 27.12 | 21.71 | 27.60 | 28.33 |

Fig. 7.   On the left: fragment of a noisy ($\sigma = 100$, PSNR 8.14 dB) grayscale *Barbara*; on the right: the corresponding fragment of the BM3D estimate (PSNR 23.49 dB) .

| **BLS-GSM** [3] | **K-SVD** [8] | **Pointwise SA-DCT [9]** | **Prelim. 3D-DFT BM3D [12]** | **Proposed BM3D** |
|---|---|---|---|---|
| PSNR 29.33 dB | PSNR 29.32 dB | PSNR 29.48 dB | PSNR 29.68 dB | PSNR 29.91 dB |



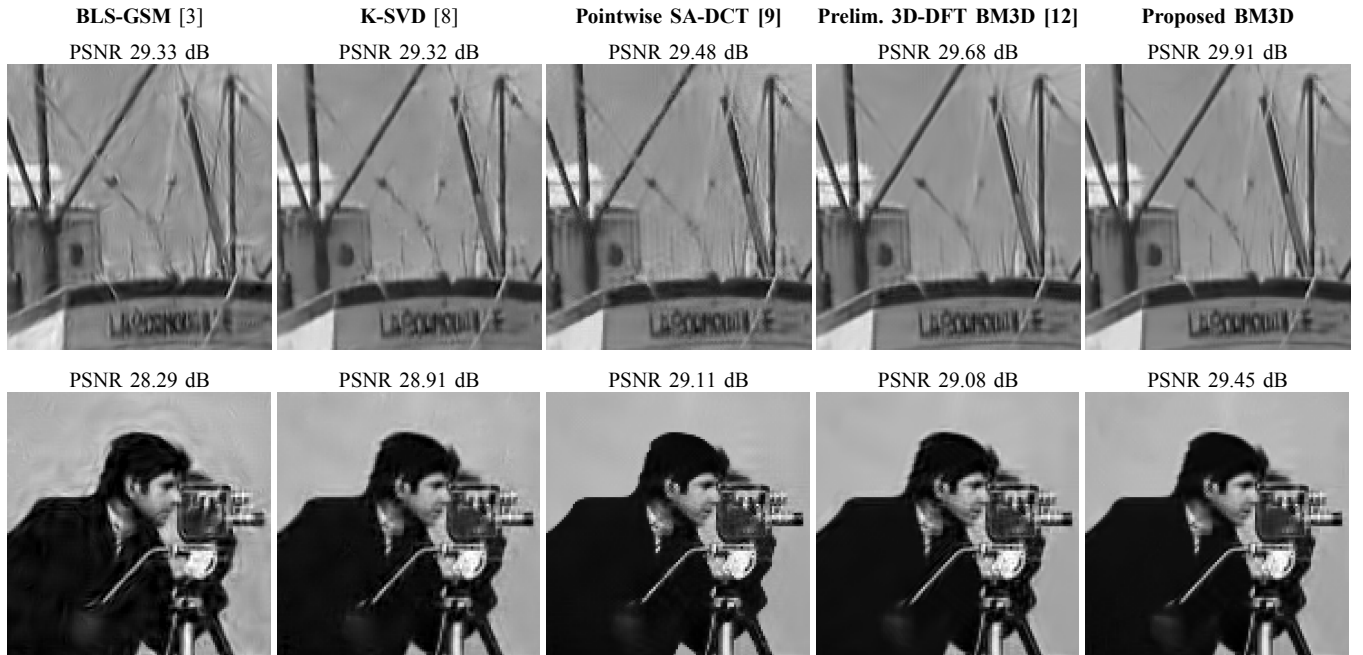| PSNR 28.29 dB | PSNR 28.91 dB | PSNR 29.11 dB | PSNR 29.08 dB | PSNR 29.45 dB |
|---|---|---|---|---|



Fig. 8.   Fragments of the grayscale *Boats* (top row) and *Cameraman* (bottom row) denoised by (from left to right): [3], [8], [9], [12], and the proposed BM3D for noise with $\sigma = 25$ (fragments of the noisy images can be seen in Figure 6).
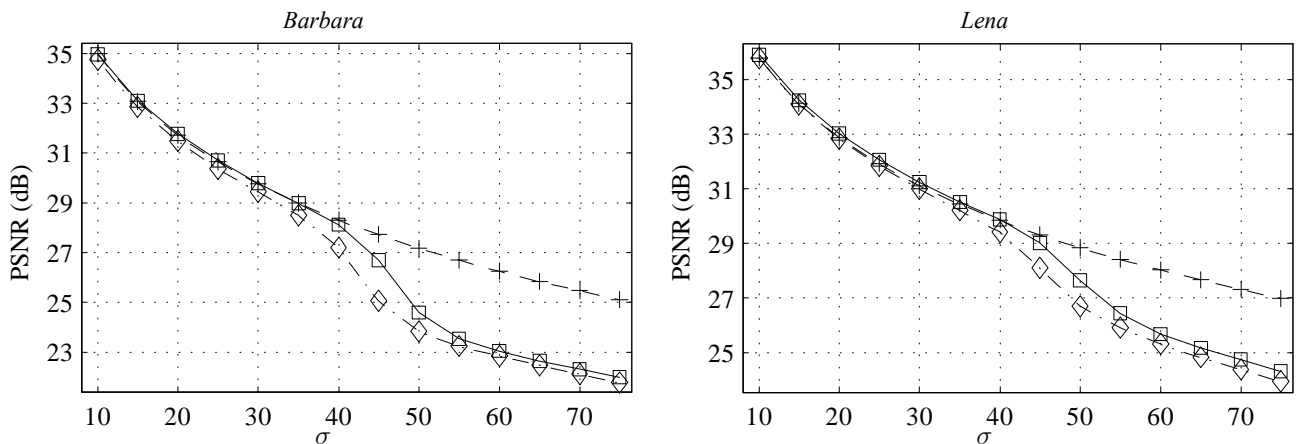


Fig. 9.   Comparison between the output PSNR corresponding to the profiles in Table I. Notation is: '◇' for Fast Profile, '□' for the Normal Profile in the case "$\sigma \leq 40$" and '+' in the case "$\sigma > 40$"; both instances of the Normal profile are shown for all considered values of $\sigma$ in the range $[10, 75]$.

TABLE V

COLOR-IMAGE DENOISING: OUTPUT-PSNR COMPARISON WITH THE TWO
STATE-OF-THE-ART RECENT METHODS [20] AND [9].

| Image | Method | Standard deviation $\sigma$ | | | |
|---|---|---|---|---|---|
| | | 10 | 15 | 20 | 25 |
| *Lena* | Proposed C-BM3D | **35.22** | **33.94** | **33.02** | **32.27** |
| $512^2$ | P.wise SA-DCT [9] | 34.95 | 33.58 | 32.61 | 31.85 |
| RGB | ProbShrink-MB [20] | 34.60 | 33.03 | 31.92 | 31.04 |
| *Peppers* | Proposed C-BM3D | **33.78** | **32.60** | **31.83** | **31.20** |
| $512^2$ | P.wise SA-DCT [9] | 33.70 | 32.42 | 31.57 | 30.90 |
| RGB | ProbShrink-MB [20] | 33.44 | 32.05 | 31.12 | 30.35 |
| *Baboon* | Proposed C-BM3D | **30.64** | **28.39** | **26.97** | **25.95** |
| $512^2$ | P.wise SA-DCT [9] | 30.62 | 28.33 | 26.89 | 25.86 |
| RGB | ProbShrink-MB [20] | 30.17 | 27.83 | 26.38 | 25.27 |
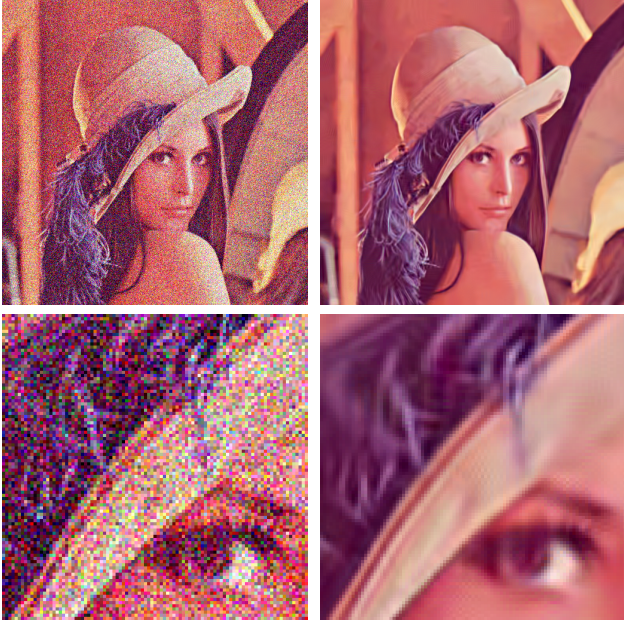


Fig. 10. Color-image denoising: on the left are a noisy *Lena* image ($\sigma = 50$, PSNR 14.15 dB) and a fragment of it; on the right are the C-BM3D estimate (PSNR 29.72 dB) and the corresponding fragment. (Color version of the figure available in the online version of the article in IEEE Xplore.)

TABLE VI

PSNR RESULTS OF THREE DIFFERENT APPROACHES TO COLOR-IMAGE
DENOISING. THE NOISE WAS ADDED IN *RGB* WITH $\sigma = 25$ AND ALL
PSNR (DB) VALUES WERE ALSO COMPUTED IN *RGB* SPACE.

| Approach to color-image denoising | *Lena* | *House* | *Peppers* |
|---|---|---|---|
| BM3D independently on the *R*, *G*, and *B* color channels | 31.44 | 32.18 | 30.93 |
| BM3D independently on each opponent color channel | 32.01 | 32.64 | 31.01 |
| C-BM3D, with grouping constraint in opponent color space | **32.27** | **33.03** | **31.20** |

images. This development has been summarized in the recent book [21].

These techniques are based on fixed-order approximations. For image processing these approximations are in practice reduced to zero and first order polynomials. It became clear that the developed neighborhood adaptivity had practically exhausted its estimation potential.

The breakthrough appears when the adaptive order local approximations are introduced. First, it was done in terms of the orthonormal transform with varying window size [22]. The hard-thresholding of the spectrum of these transforms means that some terms in the approximating series are adaptively dropped and thus the order of the model becomes data dependent [23]. The most efficient development of the idea of the adaptive order estimation in local neighborhoods was the Pointwise Shape-Adaptive DCT filter [9], where the orthonormal transform is calculated in adaptive shape neighborhoods defined by special statistical rules.

The next essential step in the development of the local approximations is presented in this paper. The spatial adaptivity is realized by selection of sets of blocks similar to a given reference one. Thus local estimates become non-local. The selected blocks are grouped in 3D arrays, jointly filtered, and aggregated at the places where they were taken from. The joint filtering of the blocks in the 3D arrays is realized by shrinkage of the spectrum items thus the idea of the order adaptive estimation is exploited again but in quite a specific way. The main advantages of this approach are the non-locality and the collaborative filtering. The latter results in effective preservation of local features in image blocks and very efficient denoising.

We wish to mention the work of a few other authors in order to clarify the context of our contribution and to state what makes it different from other similar approaches.

Since our method and the non-local estimators [10] and [11] are based on the same assumptions about the signal, it is worth comparing this class of techniques with our method. The weighted mean used in the non-local estimation corresponds to a zero-order polynomial approximation. Its effectiveness depends on an elaborate computation of adaptive weights, depending on the similarity between image patches centered at the estimated pixel and the ones used in the averaging. Our approach is different; by using a more flexible set of the basis functions (embedded in the transform), we enable order-adaptivity of the model and a more efficient exploitation of the similarity between grouped blocks. This is realized by collaborative filtering that allows for high-order estimates (not only weighted means) to be calculated for all grouped blocks.

The algorithm proposed in [8] is derived from a global optimization formulation. The image is segmented in a set of overlapping blocks and the filtering is enabled by fitting a minimum complexity model to each of these blocks. The final image estimate is obtained by fusing these models. A very good performance of the algorithm mainly follows from using a set of basis functions (dictionaries) obtained by training. In contrast, our collaborative filtering is essentially different because the model induced by hard-thresholding has low-complexity *only in relation to the group as a whole*. For

the block-wise estimates and for the image overall, the model can instead be highly complex and redundant as each block can enter in many groups and thus can participate in many collaborative estimates. This redundancy gives a very good noise attenuation and allows to avoid artifacts typical for the standard thresholding schemes. Thus, we may say that instead of some low-complexity modeling as in [8], we exploit specific overcomplete representations.

The collaborative Wiener filtering used in the second step and the aggregation of block-wise estimates using adaptive weights are major features of our approach. The Wiener filtering uses the power spectrum of the basic estimate to filter the formed groups. In result, the estimation improves significantly over the hard-thresholding used in the first step. The improvement in PSNR can be seen from Table II (by comparing the numbers in the column of "$\mathcal{T}_{2D}^{ht}$" with the numbers in any of the other two columns "$\mathcal{T}_{2D}^{wie}$" or "$\mathcal{T}_{1D}$"); one can observe that the improvement is substantial, typically greater than 0.5 dB.

The basis functions used in our algorithm are standard ones, computationally efficient, and image independent. We believe that the proposed denoising method could be improved by using more sophisticated bases such as adaptive PCA [7], or overcomplete learned dictionaries [8]. However, the computational complexity would significantly increase because these transforms are typically non-separable and do not have fast algorithms. As it is shown in the previous section, even with the currently used standard transforms, our algorithm already demonstrates better performance than both [8] and [11].

The proposed extension to color images is non-trivial because we do not apply the grayscale BM3D independently on the three luminance-chrominance channels but we impose a *grouping constraint* on both chrominances. The grouping constraint means that the grouping is done only once, in the luminance (which typically has a higher SNR than the chrominances), and exactly the same grouping is reused for collaborative filtering in both chrominances. It is worth comparing the performance of the proposed C-BM3D versus the independent application of the grayscale BM3D on the individual color channels. This is done in Table VI which shows that the C-BM3D achieves 0.2 − 0.4 dB better PSNR than the independent application of the BM3D on the opponent color channels and 0.3 − 0.8 dB better PSNR than the independent application of the BM3D on the *RGB* channels. This improvement shows the significant benefit of using the grouping constraint on the chrominances in the C-BM3D.

We note that a similar idea of filtering the chrominances using information from the luminance was exploited already in the Pointwise SA-DCT denoising method [9]. There, adaptive-shape estimation neighborhoods are determined only for $Y$ and then reused for both $U$ and $V$. The PSNR improvement (0.1 − 0.4 dB) of the proposed approach compared with [9] is consistent with the improvement between the grayscale versions of these two methods.

## VIII. CONCLUSIONS

The image modeling and estimation algorithm developed in this paper can be interpreted as a novel approach to non-local adaptive nonparametric filtering. The algorithm demonstrates state-of-the-art performance. To the best of our knowledge, the PSNR results shown in Tables III and IV are the highest for denoising additive white Gaussian noise from grayscale and color images, respectively. Furthermore, the algorithm achieves these results at reasonable computational cost and allows for effective complexity/performance trade-off, as shown in Table I.

The proposed approach can be adapted to various noise models such as additive colored noise, non-Gaussian noise, etc., by modifying the calculation of coefficients' variances in the basic and Wiener parts of the algorithm. In addition, the developed method can be modified for denoising 1D-signals and video, for image restoration, as well as for other problems that can benefit from highly sparse signal representations.

## REFERENCES

[1] L. Sendur and I. W. Selesnick, "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency," *IEEE Trans. Signal Process.*, vol. 50, no. 11, pp. 2744–2756, November 2002.

[2] A. Pizurica, W. Philips, I. Lemahieu, and M. Acheroy, "A joint inter- and intrascale statistical model for Bayesian wavelet based image denoising," *IEEE Trans. Image Process.*, vol. 11, no. 5, pp. 545–557, May 2002.

[3] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli, "Image denoising using a scale mixture of Gaussians in the wavelet domain," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, November 2003.

[4] J. Guerrero-Colon and J. Portilla, "Two-level adaptive denoising using Gaussian scale mixtures in overcomplete oriented pyramids," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, Genova, Italy, September 2005.

[5] L. Yaroslavsky, K. Egiazarian, and J. Astola, "Transform domain image restoration methods: review, comparison and interpretation," in *Proc. Nonlinear Image Process. and Pattern Analysis XII*, vol. 4304, 2001, pp. 155–169.

[6] O. Guleryuz, "Weighted overcomplete denoising," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, vol. 2, Pacific Grove, CA, USA, November 2003, pp. 1992–1996.

[7] D. Muresan and T. Parks, "Adaptive principal components and image denoising," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, September 2003.

[8] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. on Image Process.*, vol. 15, no. 12, pp. 3736–3745, December 2006.

[9] 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.

[10] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multisc. Model. Simulat.*, vol. 4, no. 2, pp. 490–530, 2005.

[11] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2866–2878, October 2006.

[12] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," in *Proc. SPIE Electronic Imaging: Algorithms and Systems V*, vol. 6064A-30, San Jose, CA, USA, January 2006.

[13] ——, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," in *IEEE Int. Conf. Image Process.*, San Antonio, Texas, September 2007, accepted.

[14] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. Berkeley Symp. Math. Stat. Probab.*, Berkeley, 1967, pp. 281–297.

[15] T. Kohonen, *Self-Organizing Maps*, 2nd ed., ser. Information Sciences. Springer, Heidelberg, 1997, vol. 30.

[16] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis.* Wiley, Chichester, 1999.

[17] A. Gersho, "On the structure of vector quantizers," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 157–166, 1982.

[18] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[19] E. Vansteenkiste, D. Van der Weken, W. Philips, and E. E. Kerre, "Perceived image quality measurement of state-of-the-art noise reduction schemes," in *Lecture Notes in Computer Science ACIVS*, vol. 4179, Antwerp, Belgium, September 2006, pp. 114–124.

[20] A. Pizurica and W. Philips, "Estimating the probability of the presence of a signal of interest in multiresolution single- and multiband image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 654–665, March 2006.

[21] V. Katkovnik, K. Egiazarian, and J. Astola, *Local Approximation Techniques in Signal and Image Process.* SPIE Press, 2006, vol. PM157.

[22] ——, "Adaptive window size image de-noising based on intersection of confidence intervals (ICI) rule," *Math. Imaging and Vision*, vol. 16, no. 3, pp. 223–235, May 2002.

[23] A. Foi and V. Katkovnik, "From local polynomial approximation to pointwise shape-adaptive transforms: an evolutionary nonparametric regression perspective," in *Proc. Int. TICSP Workshop Spectral Methods Multirate Signal Process.*, Florence, Italy, September 2006.

**Kostadin Dabov** received the M.Sc. degree in digital signal processing at the Institute of Signal Processing at Tampere University of Technology in April 2006. There he is currently pursuing a D.Tech. degree on the topic of signal restoration with locally adaptive transform-based techniques. His interests include signal enhancement and restoration, and efficient design and realization of signal processing algorithms.

**Alessandro Foi** has received the M.Sc. and the Ph.D. degrees in mathematics from Università degli Studi di Milano (Italy) in 2001 and from Politecnico di Milano in 2005, respectively. His research interests include mathematical and statistical methods for signal processing, functional analysis, and harmonic analysis. Currently, he is a researcher at the Institute of Signal Processing, Tampere University of Technology (Finland). His work focuses on spatially adaptive algorithms for denoising and deblurring of digital images and on noise modeling for digital imaging sensors.

**Vladimir Katkovnik** received the M.Sc., Ph.D., and D.Sc. degrees in technical cybernetics from the Leningrad Polytechnic Institute, Leningrad, Russia, in 1960, 1964, and 1974, respectively. From 1964 to 1991, he held the positions of Associate Professor and Professor at the Department of Mechanics and Control Processes, Leningrad Polytechnic Institute. From 1991 to 1999, he was a Professor of statistics with the Department of the University of South Africa, Pretoria. From 2001 to 2003, he was a Professor of mechatronics with the Kwangju Institute of Science and Technology, Korea. From 2000 to 2001 and since 2003 he is a Research Professor with the Institute of Signal Processing, Tampere University of Technology, Tampere, Finland. He has published seven books and more than 200 papers. His research interests include stochastic signal processing, linear and nonlinear filtering, nonparametric estimation, imaging, nonstationary systems, and time-frequency analysis.

**Karen Egiazarian** (SM'96) was born in Yerevan, Armenia, in 1959. He received the M.Sc. degree in mathematics from Yerevan State University in 1981, the Ph.D. degree in physics and mathematics from Moscow State University, Moscow, Russia, in 1986, and the D.Tech. degree from the Tampere University of Technology (TUT), Tampere, Finland, in 1994. He has been Senior Researcher with the Department of Digital Signal Processing, Institute of Information Problems and Automation, National Academy of Sciences of Armenia. Since 1996, he has been an Assistant Professor with the Institute of Signal Processing, TUT, where he is currently a Professor, leading the Transforms and Spectral Methods group. His research interests are in the areas of applied mathematics, signal processing, and digital logic.