Journal of Electronic Imaging

SPIEDigitalLibrary.org/jei

Wiener discrete cosine transform-based image filtering

Oleksiy Pogrebnyak
Vladimir V. Lukin

Wiener discrete cosine transform-based image filtering

Oleksiy Pogrebnyak
Instituto Politecnico Nacional
Centro de Investigacion en Computacion
Av. Miguel Othon Mendizabal s/n, Col. La Escalera, Del. Gustavo A. Madero
C.P. 07320 Mexico City, Mexico
E-mail: olek@pollux.cic.ipn.mx

Vladimir V. Lukin
National Aerospace University
Department 504, 17 Chkalova Street
61070 Kharkov, Ukraine

Abstract. *A classical problem of additive white (spatially uncorrelated) Gaussian noise suppression in grayscale images is considered. The main attention is paid to discrete cosine transform (DCT)-based denoising, in particular, to image processing in blocks of a limited size. The efficiency of DCT-based image filtering with hard thresholding is studied for different sizes of overlapped blocks. A multiscale approach that aggregates the outputs of DCT filters having different overlapped block sizes is proposed. Later, a two-stage denoising procedure that presumes the use of the multiscale DCT-based filtering with hard thresholding at the first stage and a multiscale Wiener DCT-based filtering at the second stage is proposed and tested. The efficiency of the proposed multiscale DCT-based filtering is compared to the state-of-the-art block-matching and three-dimensional filter. Next, the potentially reachable multiscale filtering efficiency in terms of output mean square error (MSE) is studied. The obtained results are of the same order as those obtained by Chatterjee's approach based on nonlocal patch processing. It is shown that the ideal Wiener DCT-based filter potential is usually higher when noise variance is high.* [DOI: 10.1117/1.JEI.21.4.043020]

1 Introduction

Noise is one of the main factors that degrades image quality.[1,2] In spite of considerable efforts spent on noise intensity reduction in originally acquired images, noise still remains visible and disturbing for many practical applications. There are different types of noise that can be present in images such as additive white Gaussian noise (AWGN), spatially correlated additive noise, signal-dependent and mixed noise, speckle, etc.[3–6] And there are various groups of methods for image denoising. However, researchers continue their attempts to design new, more efficient techniques for both quite general and more specific applications.

One reason is that the image processing community and customers are not satisfied by the already obtained results. Another reason is that until recently it has not been clear that there is room for further improvement of image filtering performance. Fortunately, a new approach to the estimation of potential limit output (PLO) mean square error (MSE) for grayscale (one-component) images has been put forward by Chatterjee and Milanfar.[7] This approach presumes that noise is AWGN and a noise-free image is available. Later, this approach has been further advanced[8] to allow predicting the PLO MSE without having a quite accurate corresponding noise-free image.

The results presented in Refs. 8–10 demonstrate the following: for a given image, the PLO MSE decreases if noise variance reduces. For a given noise variance, the PLO MSE can vary by several times depending upon an image. It can be easily concluded from data presented in Ref. 7 that the PLO MSE is considerably, by up to 10 times, larger for more complex structure (highly textural) images. Within the approach in Ref. 7, the PLO MSE is practically reached by modern most efficient filters for complex-structure images.

The PLO MSE in Ref. 7 has been derived within a nonlocal filtering approach. There are many techniques that belong to this family nowadays. They are based on searching for similar patches and their joint processing.[11–14] Among them, the block-matching three-dimensional (BM3D) filter[14] has been shown to be the most efficient for processing most grayscale test images[7] and component-wise denoising of color test images[10] corrupted by AWGN.

Meanwhile, the approach in Ref. 7 might not be unique for determination of PLO MSE. From the linear filtering theory, the Wiener filter is known to be the optimal in the sense of providing minimal output MSE under the condition of *a priori* known spectra of stationary signal and noise.[15] Wiener filtering being applied to processing an entire image in spatial two-dimensional (2-D) Fourier domain is not as efficient as in the case of one-dimensional (1-D) stationary signal filtering (stationarity is required for proper operation of the Wiener filter[16]), since images are nonstationary random 2-D processes. Because of this, quasi-Wiener filtering is often implemented in spatial domain locally. The widely known local statistic Lee[17] and Kuan[18] filters are good examples of such algorithms. There are also options of the Wiener filter used in other than Fourier orthogonal transforms as, e.g., wavelet,[16,19–21] DCT,[4,22,23] and others.[22] An attempt to implement a nonlocal Wiener filter in spatial domain using image "photometric similarities" is presented in Ref. 24.

Paper 12145 received Apr. 19, 2012; revised manuscript received Nov. 7, 2012; accepted for publication Nov. 12, 2012; published online Dec. 13, 2012.

0091-3286/2012/$25.00

Reference 22 compares the Wiener-based filtering efficiency for different orthogonal bases. Although this is done for the 1-D case, an important conclusion is that the DCT domain Wiener filtering approaches the best known optimal Karhunen-Loeve transform basis. This is due to very good data de-correlation and the energy compaction properties of the DCT, which are widely exploited in image and video compression.[25] Efficiency and usefulness of the local DCT commonly carried out in 8×8 pixel blocks has also been proven for image denoising applications in Refs. 26–31. Thus, below we focus just on DCT as the considered basic orthogonal transform.

In this paper, our goal is to analyze the potential of the DCT image filtering in detail including an ideal (hypothetical) case of *a priori* known global and local power spectra and a more practical case when only information on noise statistics (variance) is available. Next, we determine the potential limits of the DCT-based filtering efficiency for fully overlapping blocks of 4×4, 8×8, and 16×16 pixels within the Wiener approach and compare them to the results obtained by the Chatterjee's approach[7,24] for a wide set of standard test images. Also, we analyze the filtering efficiency of the proposed multiscale DCT-based filters and compare them to the state-of-the-art BM3D filter.

The paper is organized as follows: the image Wiener filtering principle is considered; a way on how it reduces to hard switching filter is shown in Sec. 2. Details of multiscale DCT-based filtering are presented in Sec. 3. Numerical simulation results for two proposed multiscale filters in comparison to the best known ones are presented in Sec. 4, providing wide opportunities for analysis and comparisons. A brief discussion of what else can be done in DCT-based filtering is presented in Sec. 5. Finally, the conclusions follow.

2 Image Wiener Filtering in DCT Domain

Let us consider an additive observation equation (model)

$$u(x,y) = s(x,y) + n(x,y), \tag{1}$$

where $u(x,y)$ is an observed noisy image; x, y are Cartesian coordinates; $s(x,y)$ denotes a noise-free image; and $n(x,y)$ is a white Gaussian noise not correlated with $s(x,y)$. The problem is to find an estimate of the noise-free image $\hat{s}(x,y)$ such that it minimizes MSE $E\{[s(x,y) - \hat{s}(x,y)]^2\}$, where $E\{\cdot\}$ denotes the expectation operator.

The optimal linear filter that minimizes the MSE is the well-known Wiener filter.[14] It is the solution of Wiener-Hopf equations expressed in matrix form as[14]

$$\mathbf{Rw} = \mathbf{p}, \tag{2}$$

where $\mathbf{R}$ is an autocorrelation matrix of a noisy image, $\mathbf{w}$ is a vector of Wiener filter impulse response coefficients, and $\mathbf{p}$ is a vector of cross-correlation between the noisy and noise-free images. Alternatively, the Wiener-Hopf equations can be represented as

$$\mathbf{r} * \mathbf{w} = \mathbf{p}, \tag{3}$$

where $\mathbf{r} = \mathbf{r}_s + \mathbf{r}_n$ is a vector of noisy image $u(x,y)$ auto-correlation function in the case of the additive noise model [Eq. (1)], $*$ denotes convolution operation, $\mathbf{r}_s$ is an auto-correlation function of the 2-D signal $s(x,y)$, and $\mathbf{r}_n$ is an auto-correlation function of the noise. Using the Fourier transform property for convolution and the Wiener-Khinchin theorem that relays correlation and power spectrum, one can obtain the Wiener-Hopf equation in the spectral domain given for the 2-D case as:

$$[P_s(\omega_x,\omega_y) + P_n(\omega_x,\omega_y)] \cdot H_W(\omega_x,\omega_y) = P_{us}(\omega_x,\omega_y), \tag{4}$$

where $P_s(\omega_x,\omega_y) = |\mathcal{F}\{\mathbf{r}_s\}|^2$, $P_n(\omega_x,\omega_y) = |\mathcal{F}\{\mathbf{r}_n\}|$ are power spectral densities of the noise-free image and noise, respectively; $\mathcal{F}\{\cdot\}$ denotes Fourier transform; ω_x, ω_y are spatial frequencies; $P_{us}(\omega_x,\omega_y) = |\mathcal{F}\{\mathbf{p}\}|^2$ is a cross spectrum between noisy image and noise-free image; and $H_W(\omega_x,\omega_y)$ is a 2-D frequency response of the Wiener filter. When the noise is not correlated with the image, $\mathbf{p} = \mathbf{r}_s$ and the following expression holds:

$$P_{us}(\omega_x,\omega_y) = P_s(\omega_x,\omega_y). \tag{5}$$

Thus, the Wiener filter in the spectral domain can be formulated as

$$H_W(\omega_x,\omega_y) = \frac{P_s(\omega_x,\omega_y)}{P_s(\omega_x,\omega_y) + P_n(\omega_x,\omega_y)}. \tag{6}$$

In practice, the exact power spectral densities $P_s(\omega_x,\omega_y)$, $P_n(\omega_x,\omega_y)$ are often unavailable. A more realistic case presumes the use of the estimates of spectral densities:

$$\hat{H}_W(\omega_x,\omega_y) = \frac{\hat{P}_s(\omega_x,\omega_y)}{\hat{P}_s(\omega_x,\omega_y) + \hat{P}_n(\omega_x,\omega_y)}, \tag{7}$$

where $\hat{H}_W(\omega_x,\omega_y)$ is an estimate of the frequency response of the Wiener filter and $\hat{P}_s(\omega_x,\omega_y), \hat{P}_n(\omega_x,\omega_y)$ are power spectral density estimates of the noise-free image and noise, respectively.

In the case of additive white Gaussian noise, the model for noise power spectral density is given by:

$$\hat{P}_n(\omega_x,\omega_y) = c(\omega_x,\omega_y) \cdot \sigma^2, \tag{8}$$

where σ^2 is noise variance, $c(\omega_x,\omega_y)$ is proportional to the image size, and $c(0,0) = 0$ because we assume the Gaussian noise to have zero mean. Thus, the Wiener filter formula transforms to

$$\hat{H}_W(\omega_x,\omega_y) = \frac{\hat{P}_s(\omega_x,\omega_y)}{\hat{P}_s(\omega_x,\omega_y) + c(\omega_x,\omega_y) \cdot \sigma^2}. \tag{9}$$

In our proposal, we use the cosine transform instead of the Fourier transform for spectrum calculation, i.e., $\hat{P}_s(\omega_x,\omega_y) = [S(\omega_x,\omega_y)]^2$, where $S(\omega_x,\omega_y)$ is the DCT of a noise-free image (or its fragment). Again, in practice the noise-free image is not accessible to obtain $S(\omega_x,\omega_y)$. For this reason, the estimate of image power spectral density, $\hat{P}_s(\omega_x,\omega_y)$, should be calculated using an observed noisy image. Therefore, the image data has to be prefiltered to obtain some rough estimate of a noise-free image $\hat{S}(\omega_x,\omega_y)$ and then to calculate $\hat{P}_s(\omega_x,\omega_y)$ to implement the Wiener filter [Eq. (9)].

The last expression for the Wiener filter frequency response, Eq. (9), could be simplified assigning the unit gain for all spatial frequencies where $|U(\omega_x, \omega_y)| \geq \beta\sigma$ and zero gain otherwise. This results in a hard thresholding technique[5]:

$$H_T(\omega_x, \omega_y) = \begin{cases} 1 & \text{if } |U(\omega_x, \omega_y)| \geq \beta\sigma \\ 0 & \text{otherwise} \end{cases}, \tag{10}$$

where β is a control parameter. If $S(\omega_x, \omega_y)$ is available, the decision rule can be interpreted as $|S(\omega_x, \omega_y)| \geq \beta\sigma$, $\beta = 1$ that correspond to the Wiener filter pass band cutoff at the level of -3 dB. In practice, the decision rule is based on the observed image, $|U(\omega_x, \omega_y)| \geq \beta\sigma$.

In this case, β was proven to have quasi-optimal value $\beta \approx 2.7$.[6,23,26] To confirm this, let us present some results. Figure 1(a) shows a three-component LandsatTM image (optical bands) in red-green-blue representation. AWGN has been added to all three components and they have been processed by the DCT filter component-wise (8×8 pixel blocks with full overlapping of blocks, see details in the next sections). The dependences of the output MSE for all three components are presented in Fig. 1(b) and 1(c) for noise standard deviations 7 and 10, respectively. There are obvious minima for all dependences for β slightly larger than 2.5. Since component images are quite similar (characterized by cross-correlation factor of about 0.9), all dependences are very similar. A general tendency is that optimal β shifts to larger values for less complex images and/or larger standard deviations of the noise and vice versa. Meanwhile, setting β equal to 2 or, e.g., 3.4 (i.e., 2.7 ± 0.7) instead of 2.7 leads to an MSE increase by about 10%. Thus, optimal setting (which is individual for each image and noise standard deviation) instead of the recommended quasi-optimal is able to produce output MSE which is only a few percent smaller than $\beta \approx 2.7$.

The thresholding filter [Eq. (10)] can be used as a preliminary image estimate $\hat{s}(x, y)$ for its further use to determine $\hat{S}(\omega_x, \omega_y)$ for the Wiener filter [Eq. (9)].

3 Locally Adaptive Wiener Image Filter in DCT Domain

More accurate estimates of $\hat{P}_s(\omega_x, \omega_y)$ are used for Wiener filtering, and better results in the sense of the output MSE are achieved [or, equivalently, in the sense of the peak signal-to-noise ratio defined for byte represented images as $\text{PSNR} = 10 \log_{10}(65025/\text{MSE})$]. This way, one can use local spectral estimates $\hat{P}_s$ to take into account local data activity for better noise filtering. For this purpose, the filtering may be performed within blocks of $m \times m$ pixels, and such blocks are allowed to be overlapped for better noise suppression. In this paper, we assume that the blocks are maximally (fully) overlapped, i.e., the $m \times m$ neighboring blocks have the overlapping area of $(m-1) \times m$ pixels if their upper left corner positions are shifted with respect to each other by only one pixel. In Refs. 23 and 26, it was shown that the DCT-based filtering with block overlapping reduces blocking effects and produces better output PSNR. The DCT-based denoising with full overlapping is more efficient in the sense of output MSE criterion than processing with partial overlapping or in nonoverlapped blocks.[23] Meanwhile, denoising in fully overlapped blocks takes more time. However, since DCT can be easily implemented using fast algorithms and/or specialized software or hardware, DCT-based denoising in fully overlapped blocks is fast enough.

So, for a locally adaptive Wiener DCT-based image filter we use a normalized DCT-2 transform[32] given by

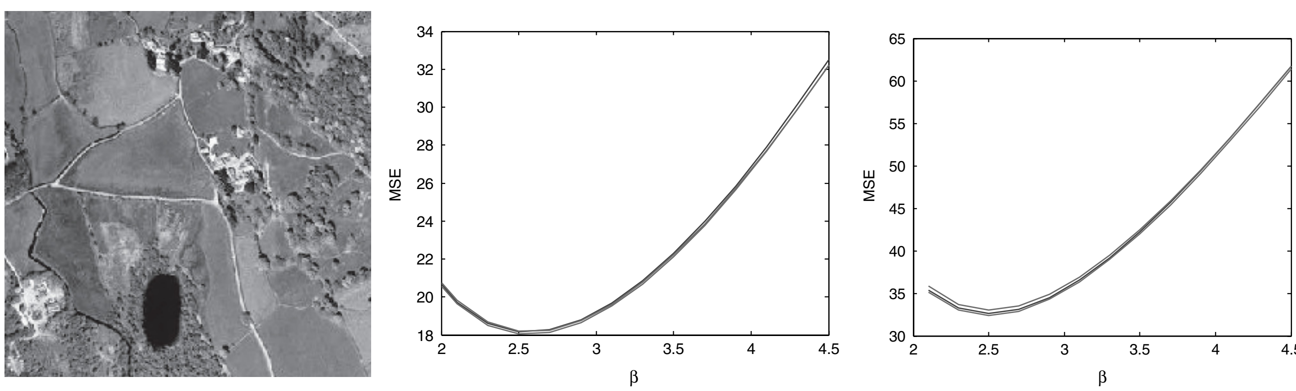
$$U^{(m)}(p, q) = \frac{\alpha(p)\alpha(q)}{m} \sum_{k=0}^{m-1} \sum_{l=0}^{m-1} u(i+k, j+l) \times \cos\left[\frac{(2k+1)p\pi}{2m}\right] \cos\left[\frac{(2l+1)q\pi}{2m}\right], \tag{11}$$

where $m \times m$ is the block size; i, j are left upper corner coordinates of the data block in the full image;

$$\alpha(x) = \begin{cases} 1, & 1 \leq x \leq m-1 \\ \frac{1}{\sqrt{2}}, & x = 0 \end{cases}.$$

The inverse transform is given by

$$u(i+k, j+l) = \frac{1}{m} \sum_{p=0}^{m-1} \sum_{q=0}^{m-1} \alpha(p)\alpha(k) U^{(m)}(p, q) \times \cos\left[\frac{(2i+1)k\pi}{2m}\right] \cos\left[\frac{(2j+1)l\pi}{2m}\right]. \tag{12}$$



(a) (b) (c)

Fig. 1 (a) Considered three-component image; (b) and (c) dependences of the output MSE on β.

Using the definition in Eq. (11), the frequency response of the local hard thresholding filter is:

$$H_T^{(m)}(p,q) = \begin{cases} 1 & \text{if } |U^{(m)}(p,q)| \geq \beta\sigma \\ 0 & \text{otherwise} \end{cases}. \tag{13}$$

The filtered image block is then obtained taking the inverse transform as

$$\hat{s}_T^{(m)}(i+k,j+l) = \frac{1}{m}\sum_{p=0}^{m-1}\sum_{q=0}^{m-1}\alpha(p)\alpha(k)U^{(m)}(p,q) \times H_T^{(m)}(p,q)\cos\left[\frac{(2i+1)k\pi}{2m}\right] \times \cos\left[\frac{(2j+1)l\pi}{2m}\right]. \tag{14}$$

Note that, opposite to scanning window filtering, the filtered values are obtained simultaneously for all pixels of a given block. And then, if processing with block overlapping is applied, these filtered values must be aggregated as described below.

Next, we propose to use the estimate in Eq. (14) to determine the local power spectrum $\hat{P}_s(p,q)$ as

$$\hat{P}_s^{(m)}(p,q) = \left\{\frac{\alpha(p)\alpha(q)}{m}\sum_{k=0}^{m-1}\sum_{l=0}^{m-1}\hat{s}_T^{(m)}[i+k,j+l] \times \cos\left[\frac{(2k+1)p\pi}{2m}\right]\cos\left[\frac{(2l+1)q\pi}{2m}\right]\right\}^2. \tag{15}$$

Using Eq. (15), the frequency response of the local Wiener DCT-based image filter can be formulated as

$$\hat{H}_W^{(m)}(p,q) = \frac{\hat{P}_s^{(m)}(p,q)}{\hat{P}_s^{(m)}(p,q) + c^{(m)}(p,q)\cdot\sigma^2}, \tag{16}$$

where

$$c^{(m)}(p,q) = \begin{cases} 0, & \text{if } p=q=0 \\ \frac{1}{m} & \text{otherwise} \end{cases}.$$

The filtered image block is obtained taking the inverse transform as

$$\hat{s}_W^{(m)}(i+k,j+l) = \frac{1}{m}\sum_{p=0}^{m-1}\sum_{q=0}^{m-1}\alpha(p)\alpha(k)U^{(m)}(p,q)\hat{H}_W^{(m)} \times (p,q)\cos\left[\frac{(2i+1)k\pi}{2m}\right]\cos\left[\frac{(2j+1)l\pi}{2m}\right]. \tag{17}$$

On the other hand, with the overlapping of the filtered blocks in Eq. (14), Eq. (17) results in a high redundancy of the filtered data that has to be aggregated to produce the filtered image $\hat{s}(i,j)$. The aggregation can be performed by averaging the block pixels where the overlapping occurs. It can also be performed using some weighting as proposed in Ref. 14, or using weighted least square patch averaging. However, we have determined by simulations that this simple mean calculation for block data aggregation

$$\hat{s}(i,j) = \sum_{q=1}^{Q(i,j)}\frac{\hat{s}_{\text{local}}^{(m)}(i,j,q)}{Q^{(m)}(i,j)} \tag{18}$$

produces appropriately good results where $\hat{s}_{\text{local}}^{(m)}(i,j,q)$ are i,j'th pixel of q'th overlapped block in Eq. (14) or Eq. (17) of size m, $Q^{(m)}(i,j)$ denotes the number of overlapping blocks in the i, j'th pixel. Note that filtering efficiency might be slightly worse for pixels near image edges since for these pixels a smaller number of filtered values from processed overlapped blocks is aggregated (for example, only one for four image corner pixels).

Next, we have found by simulations that the aggregation of the overlapped blocks of different size might further improve noise suppression. To this end, at each pixel position, different values of m in Eqs. (11), (12), (14), and (17) are used and then the processed overlapped blocks of different size are aggregated using some weighting. In particular, we have determined that the following weighting produces good results for different images and different noise levels:

$$\hat{s}(i,j) = \sum_{q=1}^{Q(i,j)}\frac{0.15\hat{s}_{\text{local}}^{(4)}(i,j,q) + \hat{s}_{\text{local}}^{(8)}(i,j,q) + 0.5\hat{s}_{\text{local}}^{(16)}(i,j,q)}{0.15Q^{(4)}(i,j) + Q^{(8)}(i,j) + 0.5Q^{(16)}(i,j)}, \tag{19}$$

where $Q^{(m)}(i,j)$ is the number of overlapped blocks of size $m \times m$. This approach will be further denoted as a multiscale DCT-based filter (MDF). The recommended weight setting in Eq. (19) is based on the results presented in the next section.

4 Simulation Results

The simulations have been performed using a wide set of standard grayscale test images[33] shown in Fig. 2, all of size 512×512 pixels. This allows obtaining quite full imagination on properties and performance of different filtering algorithms and approaches considered in this paper. Noise variance (standard deviation) has been varied in a very wide range as well. Despite the noise standard deviation values of the order 20 ... 35 for grayscale images of 8-bit representation it is almost impossible to meet, in practice, the corresponding data often presented in literature dealing with filter efficiency analysis and comparisons.[7,12,14] Thus, we have decided to obtain and present such data for the considered techniques.

Fig. 2 Test images: Lena, Boats, F-16, Man, Stream & bridge, Aerial, Baboon, Sailboat, Elaine, Couple, Tiffany, and Peppers.

Table 1 Performance (in terms of the output PSNR, in dB) of the standard DCT-based filtering techniques [Eqs. (9) and (10)] and the ideal Wiener filtering that all operate over entire image transformed data.

Image	σ	DCT hard thresholding	Wiener filtering	Ideal Wiener filtering
Lena	2	39.797	39.916	44.936
	5	34.089	34.247	39.398
	10	30.472	30.671	35.795
	15	28.44	28.676	33.895
	20	27.025	27.294	32.631
	25	25.931	26.23	31.697
	30	25.032	25.364	30.96
	35	24.251	24.61	30.356
Boats	2	39.883	39.976	44.421
	5	33.213	33.354	38.468
	10	29.112	29.289	34.56
	15	26.974	27.179	32.514
	20	25.57	25.801	31.167
	25	24.506	24.76	30.18
	30	23.631	23.908	29.411
	35	22.929	23.228	28.786
F-16	2	40.309	40.421	45.08
	5	34.242	34.39	39.353
	10	30.205	30.389	35.491
	15	27.984	28.198	33.409
	20	26.41	26.651	32.011
	25	25.26	25.524	30.971
	30	24.31	24.596	30.15
	35	23.456	23.761	29.474
Man	2	39.497	39.585	44.175
	5	32.729	32.877	38.24
	10	28.943	29.126	34.445
	15	27.038	27.248	32.498

Table 1 (*Continued*).

Image	σ	DCT hard thresholding	Wiener filtering	Ideal Wiener filtering
Man	20	25.756	25.993	31.228
	25	24.785	25.047	30.3
	30	23.981	24.268	29.575
	35	23.267	23.578	28.983
Stream & bridge	2	39.808	39.843	43.373
	5	31.533	31.654	36.896
	10	26.940	27.103	32.704
	15	24.886	25.069	30.568
	20	23.608	23.809	29.194
	25	22.704	22.922	28.206
	30	21.974	22.211	27.448
	35	21.385	21.64	26.839
Aerial	2	39.789	39.858	43.801
	5	32.385	32.508	37.489
	10	27.898	28.053	33.297
	15	25.601	25.776	31.082
	20	24.069	24.262	29.617
	25	22.933	23.145	28.541
	30	22.053	22.282	27.7
	35	21.317	21.562	27.016
Baboon	2	40.105	40.124	43.148
	5	31.524	31.617	36.405
	10	26.244	26.387	31.942
	15	23.778	23.946	29.649
	20	22.313	22.499	28.175
	25	21.342	21.545	27.122
	30	20.623	20.841	26.319
	35	20.058	20.291	25.682
Sailboat	2	39.479	39.566	44.088
	5	32.724	32.868	38.093

Table 1 (*Continued*).

Image	σ	DCT hard thresholding	Wiener filtering	Ideal Wiener filtering
Sailboat	10	28.889	29.061	34.208
	15	26.823	27.019	32.167
	20	25.406	25.626	30.811
	25	24.332	24.572	29.806
	30	23.46	23.722	29.014
	35	22.701	22.983	28.364
Elaine	2	39.499	39.627	44.959
	5	34.165	34.339	39.636
	10	31.139	31.356	36.325
	15	29.333	29.591	34.604
	20	28	28.298	33.448
	25	26.877	27.21	32.58
	30	25.98	26.35	31.885
	35	25.118	25.519	31.307
Couple	2	39.499	39.571	43.864
	5	32.193	32.332	37.751
	10	28.245	28.421	33.854
	15	26.375	26.575	31.869
	20	25.142	25.367	30.585
	25	24.225	24.472	29.657
	30	23.472	23.743	28.938
	35	22.838	23.136	28.358
Tiffany	2	39.443	39.553	44.626
	5	33.458	33.62	39.084
	10	30.288	30.49	35.637
	15	28.609	28.849	33.883
	20	27.394	27.67	32.737
	25	26.376	26.69	31.9
	30	25.532	25.879	31.244
	35	24.765	25.14	30.709

Table 1 (*Continued*).

Image	σ	DCT hard thresholding	Wiener filtering	Ideal Wiener filtering
Peppers	2	39.475	39.589	44.646
	5	33.608	33.772	39.064
	10	30.239	30.44	35.51
	15	28.345	28.579	33.641
	20	26.981	27.241	32.388
	25	25.862	26.15	31.451
	30	24.925	25.241	30.706
	35	24.104	24.445	30.089

4.1 *DCT Domain Hard Thresholding and Wiener Denoising*

Let us start by applying filtering to the entire image: the DCT hard thresholding [Eq. (13)], practical Wiener filtering [with spectrum estimation from DCT filtered image; Eq. (16)], and the ideal Wiener (when P_s, P_n are both known). The obtained results are presented in Table 1.

As can be easily expected, the output PSNR decreases if noise standard deviation becomes larger (this tendency is observed for any filtering approach). However, output PSNR values differ a lot. For example, for the noise standard deviation equal to 10, the DCT-based filtering with hard thresholding (the quasi-optimal $\beta \approx 2.7$ has been used for all images and values of noise standard deviation) produces output PSNR ranging from 31.14 dB for the simple structure Elaine image to 26.24 dB for the complex structure Baboon image. Similarly, the output PSNR for the ideal Wiener filter ranges from 36.33 to 31.94 dB (again, for the test images Elaine and Baboon, respectively).

A more detailed analysis shows that the output PSNR values for the ideal Wiener filter are usually by $3 \ldots 7$ dB larger than for the DCT-based filter with hard thresholding. The difference slightly increases if the noise standard deviation becomes larger. The difference is smaller for the test images with more complex structure such as Baboon and Stream & bridge.

The two-stage procedure of practical Wiener filtering produces intermediate results which are considerably closer to the outputs of the DCT-based filter with hard thresholding than to the ideal Wiener filter. The resulting PSNR for the practical Wiener filter can be up to 0.4 dB better than for the DCT-based filtering with hard thresholding. This means that the estimates of the power spectrum $\hat{P}_s(\omega_x, \omega_y)$ are not accurate enough. Note that the largest improvement for the practical Wiener filter occurs for the test images with quite simple structure and if the noise variance is large.

4.2 *Block-Based Denoising*

As it has been mentioned in the Introduction, images are 2-D nonstationary processes for which local spatial spectra shapes differ considerably from spatial spectra shapes for the corresponding entire images. Although 8×8 blocks are usually employed in the DCT-based filtering, we have considered the question of block size selection in more detail. For this purpose, the output PSNR values have been obtained for three sizes of m, namely 4, 8, and 16 taking into account that in such cases the DCT-based filtering can be carried out faster than for other block sizes (e.g., $m = 11$) that are, in general, also possible. The obtained results are presented in Table 2. As before, the results are given for the DCT-based filtering with hard thresholding, the practical (two-stage) Wiener filtering [Eq. (17)], and the ideal Wiener filtering. Besides, we present results for the lower bound of filtering efficiency obtained according to Ref. 7 using the software tool offered by the authors[34] (according to the recommendations in Ref. 7, the selected number of clusters $c = 5$ with the patch size $p_s z = 11$). The following results are expressed not in output MSE as it is produced by the software but in terms of PSNR for the convenience of comparisons. The same test image set is used and the AWGN with the same values of the standard deviation have been simulated.

The first observation that follows from comparison of the corresponding data in Tables 1 and 2 is that the image block-wise filtering produces considerably better results than the image filtering with DCT applied to the entire image. The output values for the block-wise version of the DCT-based filtering with hard thresholding are by $3 \ldots 4$ dB better than the entire image counterpart. This once more confirms expedience of the image local processing approach (with block overlapping). Similar observations hold for the practical and ideal Wiener filters.

As is seen, the block size $m \times m$ has sufficient impact on the DCT-based filter performance. The results for $m = 4$ are worse than for $m = 8$ or 16 in practically all cases. The only exceptions are the results for the test image Stream & bridge for small standard noise deviations where PSNR for $m = 4$ is slightly better than for $m = 16$. Meanwhile, the PSNR values for $m = 8$ and $m = 16$ usually do not differ a lot between each other, and simulations for $m = 32$ revealed the filtering efficiency reduction in comparison to $m = 16$. The general tendency is the following: $m = 16$ is a better choice if the noise standard deviation is larger and a processed image has a simpler structure.

We use the terms "simple structure" and "complex structure" images. Intuitively these terms are clear where the latter relates to more textural images. Unfortunately, until now there is no commonly accepted metric for image complexity.

The practical Wiener filter [Eq. (17)] again produces performance improvement compared to the DCT-based processing with hard thresholding. Due to applying the Wiener filter at the second stage, the output PSNR can be increased by up to 0.5 dB. We would like to stress here that the practical Wiener filtering can be performed in a pipeline manner, where the second stage processing is applied when the necessary output data of the DCT-based thresholding is obtained. Thus, although computation expenses are increased for the proposed two-stage procedure compared to the standard DCT-based denoising, the two-stage filtering is still considerably faster than most efficient denoising techniques that search for similar blocks (patches), and is usually time consuming.

The ideal Wiener filter again produces the output PSNR values that are by $3 \ldots 4$ dB larger than those corresponding to practically implementable methods. Note that for the ideal Wiener filter the best results are produced for $m = 16$ and the PLO PSNR for $m = 16$ can be by almost 0.8 dB better than for $m = 8$.

It is interesting to compare these results (that can be considered as PLO PSNR) to the corresponding data produced by the Chatterjee's approach.[7] Such comparisons can be easily made by considering, e.g., the data in the last (rightmost) two columns of Table 2 (the best attainable values of PLO PSNR are marked bold). The PLO PSNR for the Chatterjee's approach can be by almost 5 dB better (this takes place for simple structure images corrupted by AWGN with small standard deviation). Meanwhile, for complex structure images such as Baboon and Stream & bridge, the PLO PSNR for the Chatterjee's approach can be by almost 4 dB smaller than for the ideal Wiener filter. For images of middle complexity (as, e.g., Boat), the Chatterjee's approach produces larger PLO PSNR for small standard noise deviations than the ideal Wiener filter and vice versa. One possible explanation of this effect can be that it is a more difficult task to find similar patches and to take advantages of non-local processing for images of more complex structure and under condition where noise is intensive (has large variance).

The results presented in Table 2 also confirm one observation earlier emphasized in Ref. 9. The output PSNR for the DCT-based filtering with hard thresholding is quite close to the Chatterjee's limit[7] for the complex structure images corrupted by intensive noise (see, e.g., data for the test images Baboon and Stream & bridge for the noise standard deviation equal to 10 and larger). The difference is smaller than 1 dB. Meanwhile, there is room for efficiency improvement for simpler structure images if the noise standard deviation is not large.

4.3 *Comparison to the State-of-the-Art*

It becomes interesting to compare the performance of the proposed DCT-based filters, MDF, and two-stage Wiener MDF with the state-of-the-art BM3D filter. The data which allows carrying out such comparison are represented in Table 3. First of all, the presented PSNR values for a given image and noise standard deviation are quite close (the best results are marked bold). They differ by not more than 1 dB (this happens for simple-structure images corrupted by AWGN with large variance values, see data for the image Lena, $\sigma = 35$). The BM3D filter performs better for some test images while the two-stage Wiener filter is better for others. It is difficult to establish some obvious performance dependence of these filters on image complexity. For two simple-structure images such as Lena and Elaine, BM3D results are better for Lena and the two-stage Wiener produces, on average, better results for Elaine. Similarly, for two complex structure test images, Baboon and Stream & bridge, the two-stage Wiener filter is better for the test image Stream & bridge and vice versa.

Setting the weights in Eq. (19), we have taken into account that DCT-based denoising with 8×8 blocks usually produces not worse filtering than with 16×16 blocks but fewer artifacts are observed in neighborhoods of high-contrast edges and small-sized objects. In turn, denoising in 4×4 block is less efficient than for larger sizes of blocks.

Table 2 Output PSNR (in dB) of the DCT-based image filters [Eqs. (14), (17), and (18)] in comparison to the noise suppression bound calculated according to Ref. 7 (5 clusters were used with the patch size 11).

		DCT with hard thresholding			Wiener filtering			Ideal Wiener filtering			
Image	σ	$m = 4$	$m = 8$	$m = 16$	$m = 4$	$m = 8$	$m = 16$	$m = 4$	$m = 8$	$m = 16$	**PSNR bound**[7]
Lena	2	43.196	43.329	43.327	43.379	43.478	43.483	47.225	47.687	47.778	**52.346**
	5	38.299	38.501	38.446	38.326	38.534	38.465	42.041	42.787	42.923	**45.267**
	10	34.956	35.39	35.372	34.959	35.489	35.474	38.22	39.334	39.552	**40.561**
	15	32.89	33.501	33.52	32.885	33.677	33.706	35.915	37.37	37.691	**38.063**
	20	31.352	32.114	32.164	31.331	32.353	32.424	34.211	35.976	**36.406**	36.402
	25	30.1	31.004	31.094	30.065	31.301	31.421	32.839	34.88	**35.422**	35.179
	30	29.042	30.088	30.216	28.99	30.431	30.602	31.683	33.97	**34.623**	34.222
	35	28.106	29.29	29.467	28.042	29.678	29.909	30.68	33.201	**33.961**	33.441
Boats	2	42.764	43.02	43.025	42.942	43.134	43.14	46.255	46.636	46.63	**49.616**
	5	36.904	37.085	36.981	36.962	37.16	37.072	40.901	41.469	41.46	**42.523**
	10	33.377	33.543	33.368	33.432	33.646	33.461	37.059	37.864	**37.901**	37.741
	15	31.332	31.576	31.387	31.417	31.748	31.546	34.81	35.85	**35.952**	35.190
	20	29.851	30.18	30.004	29.94	30.402	30.208	33.183	34.448	**34.625**	33.498
	25	28.667	29.099	28.946	28.756	29.36	29.18	31.891	33.369	**33.623**	32.255
	30	27.658	28.221	28.099	27.755	28.51	28.359	30.81	32.487	**32.82**	31.285
	35	26.756	27.479	27.396	26.864	27.793	27.681	29.876	31.738	**32.149**	30.497
F-16	2	44.357	44.523	44.458	44.47	44.611	44.558	47.914	48.374	48.378	**49.815**
	5	39.246	39.358	39.178	39.264	39.446	39.271	42.549	43.242	**43.246**	42.924
	10	35.45	35.676	35.442	35.461	35.857	35.631	38.521	39.566	**39.625**	38.300
	15	33.121	33.497	33.271	33.14	33.745	33.53	36.084	37.454	**37.599**	35.823
	20	31.418	31.937	31.744	31.441	32.239	32.056	34.29	35.953	**36.193**	34.161
	25	30.064	30.732	30.583	30.088	31.077	30.939	32.853	34.778	**35.119**	32.925
	30	28.928	29.753	29.641	28.953	30.133	30.044	31.648	33.806	**34.249**	31.949
	35	27.95	28.916	28.862	27.967	29.336	29.305	30.607	32.973	**33.518**	31.146
Man	2	43.364	43.373	43.211	43.485	43.452	43.283	46.611	46.806	46.663	**49.059**
	5	37.448	37.436	37.12	37.566	37.556	37.249	41.151	41.554	**41.432**	**41.731**
	10	33.439	33.44	33.119	33.565	33.626	33.282	37.26	37.946	**37.879**	36.945
	15	31.284	31.328	31.049	31.396	31.535	31.215	34.989	35.929	**35.931**	34.525
	20	29.81	29.933	29.698	29.905	30.152	29.872	33.346	34.522	**34.599**	32.968

Table 2 (*Continued*).

		DCT with hard thresholding			Wiener filtering			Ideal Wiener filtering			
Image	σ	$m = 4$	$m = 8$	$m = 16$	$m = 4$	$m = 8$	$m = 16$	$m = 4$	$m = 8$	$m = 16$	PSNR bound[7]
Man	25	28.704	28.906	28.707	28.767	29.141	28.898	32.041	33.436	**33.591**	31.844
	30	27.79	28.102	27.929	27.825	28.357	28.145	30.95	32.548	**32.783**	30.973
	35	26.981	27.431	27.292	27	27.712	27.538	30.01	31.795	**32.111**	30.269
Stream & bridge	2	42.489	42.544	42.472	42.625	42.6	42.519	44.923	45.017	**44.952**	44.448
	5	35.489	35.518	35.368	35.671	35.647	35.493	39.044	39.298	**39.262**	36.914
	10	30.774	30.794	30.637	30.999	31.004	30.828	35.056	35.51	**35.521**	31.899
	15	28.399	28.426	28.295	28.614	28.634	28.466	32.841	33.456	**33.509**	29.421
	20	26.928	26.945	26.845	27.112	27.134	26.987	31.291	32.049	**32.143**	27.885
	25	25.897	25.911	25.837	26.049	26.084	25.959	30.09	30.98	**31.116**	26.813
	30	25.099	25.136	25.077	25.228	25.298	25.19	29.104	30.119	**30.298**	26.008
	35	24.443	24.53	24.478	24.552	24.687	24.587	28.266	29.398	**29.62**	25.371
Aerial	2	43.299	43.239	42.913	43.345	43.223	42.935	45.82	45.883	**45.622**	45.471
	5	36.777	36.641	36.167	36.824	36.695	36.27	39.994	40.236	**39.985**	38.169
	10	32.25	32.156	31.704	32.353	32.3	31.848	35.902	36.361	**36.155**	33.305
	15	29.759	29.737	29.362	29.914	29.933	29.526	33.575	34.217	**34.066**	30.781
	20	28.071	28.112	27.805	28.252	28.342	27.991	31.927	32.737	**32.645**	29.130
	25	26.819	26.902	26.655	27.003	27.155	26.858	30.64	31.608	**31.575**	27.926
	30	25.84	25.954	25.75	26.015	26.22	25.966	29.577	30.694	**30.722**	26.991
	35	25.031	25.179	25.007	25.193	25.454	25.234	28.667	29.926	**30.015**	26.234

		DCT thresholding			Wiener filtering			Ideal Wiener filtering			
Image	σ	$m = 4$	$m = 8$	$m = 16$	$m = 4$	$m = 8$	$m = 16$	$m = 4$	$m = 8$	$m = 16$	PSNR bound[7]
Baboon	2	42.151	42.302	42.34	42.319	42.38	42.392	44.396	44.603	**44.648**	44.137
	5	34.933	35.095	35.111	35.125	35.225	35.23	38.339	38.7	**38.782**	36.472
	10	30.198	30.356	30.347	30.397	30.523	30.499	34.243	34.774	**34.897**	31.186
	15	27.685	27.874	27.879	27.907	28.08	28.055	31.997	32.67	**32.829**	28.466
	20	26.027	26.248	26.274	26.249	26.471	26.46	30.443	31.248	**31.444**	26.745
	25	24.837	25.065	25.118	25.037	25.292	25.302	29.248	30.178	**30.411**	25.539
	30	23.93	24.168	24.237	24.101	24.383	24.412	28.272	29.321	**29.592**	24.640
	35	23.213	23.458	23.537	23.349	23.655	23.703	27.444	28.605	**28.916**	23.942

Table 2 (*Continued*).

Image	σ	DCT thresholding			Wiener filtering			Ideal Wiener filtering			PSNR bound[7]
		$m=4$	$m=8$	$m=16$	$m=4$	$m=8$	$m=16$	$m=4$	$m=8$	$m=16$	
Sailboat	2	42.596	42.824	42.882	42.805	42.936	42.958	45.8	46.083	46.112	**46.616**
	5	36.16	36.302	36.291	36.296	36.45	36.469	40.305	40.795	**40.831**	39.417
	10	32.568	32.631	32.462	32.632	32.714	32.541	36.469	37.172	**37.222**	34.794
	15	30.656	30.782	30.573	30.717	30.907	30.672	34.261	35.151	**35.227**	32.439
	20	29.269	29.486	29.281	29.339	29.659	29.428	32.678	33.748	**33.861**	30.897
	25	28.167	28.461	28.276	28.234	28.681	28.473	31.427	32.671	**32.827**	29.761
	30	27.225	27.609	27.457	27.295	27.872	27.696	30.386	31.795	**31.998**	28.868
	35	26.379	26.894	26.761	26.464	27.189	27.039	29.489	31.052	**31.306**	28.136
Elaine	2	42.385	42.688	42.92	42.628	42.81	42.992	45.944	46.403	46.732	**54.793**
	5	35.907	36.275	36.737	36.095	36.485	36.951	40.782	41.529	41.946	**47.596**
	10	32.92	33.18	33.481	32.872	33.148	33.483	37.186	38.198	38.643	**42.807**
	15	31.621	31.938	32.089	31.521	31.927	32.055	35.076	36.332	36.808	**40.294**
	20	30.637	31.105	31.201	30.508	31.161	31.235	33.535	35.043	35.563	**38.561**
	25	29.756	30.416	30.496	29.603	30.545	30.624	32.294	34.056	34.631	**37.304**
	30	28.933	29.811	29.885	28.765	30.003	30.11	31.242	33.251	33.891	**36.316**
	35	28.155	29.251	29.331	27.975	29.505	29.652	30.32	32.58	33.292	**35.510**
Couple	2	42.725	42.868	42.84	42.918	43.004	42.966	46.984	47.256	47.165	**49.022**
	5	37.076	37.147	36.963	37.178	37.234	37.046	41.597	42.078	42.004	**50.355**
	10	33.323	33.463	33.25	33.429	33.605	33.377	37.628	38.411	38.406	**42.740**
	15	31.131	31.389	31.216	31.261	31.585	31.38	35.267	36.329	36.405	**37.514**
	20	29.573	29.942	29.821	29.706	30.179	30.022	33.546	34.863	**35.024**	**34.828**
	25	28.346	28.843	28.767	28.482	29.108	29	32.177	33.725	**33.972**	33.116
	30	27.34	27.959	27.915	27.469	28.251	28.183	31.033	32.791	**33.124**	31.897
	35	26.472	27.21	27.201	26.594	27.533	27.503	30.047	31.997	**32.413**	30.971
Tiffany	2	43.468	43.583	43.544	43.63	43.699	43.656	47.414	47.848	47.864	**54.471**
	5	38.397	38.563	38.41	38.484	38.668	38.518	42.258	42.992	43.091	**47.068**
	10	34.896	35.191	35.096	34.947	35.353	35.24	38.451	39.595	39.806	**42.125**
	15	32.899	33.343	33.323	32.912	33.537	33.485	36.134	37.658	37.986	**39.581**
	20	31.437	32.073	32.124	31.429	32.308	32.327	34.407	36.281	36.73	**37.937**

Table 2 (*Continued*).

Image	σ	DCT thresholding $m=4$	DCT thresholding $m=8$	DCT thresholding $m=16$	Wiener filtering $m=4$	Wiener filtering $m=8$	Wiener filtering $m=16$	Ideal Wiener filtering $m=4$	Ideal Wiener filtering $m=8$	Ideal Wiener filtering $m=16$	PSNR bound[7]
Tiffany	25	30.257	31.091	31.199	30.231	31.379	31.472	33.007	35.197	35.772	**36.750**
	30	29.235	30.27	30.438	29.197	30.615	30.793	31.821	34.295	34.997	**35.833**
	35	28.329	29.562	29.781	28.275	29.962	30.216	30.786	33.516	34.345	**35.091**
Peppers	2	42.67	42.902	42.985	42.917	43.097	43.149	46.734	47.143	47.204	**52.776**
	5	37.309	37.415	37.384	37.345	37.465	37.464	41.634	42.306	42.408	**45.475**
	10	34.471	34.653	34.477	34.419	34.679	34.484	37.932	38.906	39.058	**40.663**
	15	32.706	33.112	32.928	32.649	33.217	33.016	35.724	36.991	37.206	**38.161**
	20	31.259	31.936	31.781	31.22	32.115	31.958	34.096	35.646	35.937	**36.512**
	25	30.033	30.951	30.844	30.002	31.202	31.103	32.783	34.599	34.971	**35.301**
	30	28.969	30.101	30.047	28.933	30.414	30.382	31.673	33.732	34.19	**34.353**
	35	28.024	29.345	29.347	27.984	29.717	29.756	30.706	32.986	33.531	**33.579**

Also, note that the DCT-based processing in blocks of different size can be carried out in parallel that allows diminishing processing time.

Figure 3 illustrates filtering efficiency for a fragment of the test image "Lena." As is seen, noise removal is efficient and edge/detail preservation is good for both output images. Figure 4 presents an example of processing the test image "Baboon" by the proposed Wiener filter in comparison to the state-of-the art BM3D filter. The BM3D filter suppresses noise better in "flat" (homogeneous image) regions while the proposed filter preserves better texture and details; the filtered image in this case has a more natural appearance.

5 Discussion

It is worth briefly discussing here the mechanism of DCT-based denoising with hard thresholding. Noise is removed in DCT-components of a block for which $|U(p,q)<\beta\sigma|$ (although hard thresholding operation simultaneously introduces distortions in the corresponding signal components). Meanwhile, noise is preserved in the components when $|U(p,q)\geq\beta\sigma|$. Therefore, noise reduction should increase if the number of DCT coefficient with $|U(p,q)<\beta\sigma|$ is larger.

All simulation results presented above for the DCT-based denoising have been obtained for hard thresholding with the fixed $\beta\approx 2.7$ in Eq. (13). However, as has been mentioned above, such threshold setting is quasi-optimal. Let us demonstrate this by several examples. We have selected eight test images of different complexity widely used in image processing applications. For three values of noise standard deviation (5, 10, 15), the optimal values β_{opt} that provide maximal output PSNR have been determined. They are presented in Table 4. Besides, we have determined two probabilities: $P_{2.7\sigma}$ is the probability that DCT coefficient absolute values do not exceed 2σ and $P_{2.7\sigma}$ is the probability that DCT coefficient absolute values are larger than 2.7σ. One more characteristic of filtering efficiency has been determined: the ratio $\text{MSE}_{\text{out}}/\sigma^2$, where MSE_{out} is output MSE after denoising. The obtained data are presented in Table 4. The test images are put in such order that $P_{2\sigma}$ in the fourth column increases.

The first observation is that the probabilities $P_{2\sigma}$ and $P_{2.7\sigma}$ are highly correlated. If $P_{2\sigma}$ is smaller, then $P_{2.7\sigma}$ is usually larger. The second observation is that the values $P_{2\sigma}$ are smaller and $P_{2.7\sigma}$ are larger for more complex-structure images and smaller noise variance values. This is clear since for more complex-structure images the DCT coefficients for noise-free image have wider distribution. The third observation is that β_{opt} increases if image complexity reduces and/or noise variance becomes larger. β_{opt} varies from 2.3 to 2.8 where for most typical practical situations β_{opt} is within the limits from 2.6 to 2.7.

It seems that if $P_{2.7\sigma}$ is preliminary determined for a given image under a condition of exactly known noise variance, it can prove more careful threshold setting for providing certain benefits of filtering efficiency. Such a strategy can be treated as image/variance adaptive threshold setting. However, in our opinion, the benefits of this strategy are too small to use in practice. A more reasonable way seems to use locally adaptive setting of the thresholds, but currently we are unable to propose an algorithm to do this.

The data presented in Table 4 show that for noisy images their complexity (or, more strictly saying, complexity of image denoising task) can be indirectly characterized by the parameter $P_{2.7\sigma}$. Filtering is more efficient (smaller $\text{MSE}_{\text{out}}/\sigma^2$ are provided) if $P_{2.7\sigma}$ is smaller. Note that $\text{MSE}_{\text{out}}/\sigma^2$ can vary from 0.78 (less than 1 dB increase of output PSNR compared to input PSNR) to 0.13 and even

Table 3 Performance (PSNR, in dB) of the proposed image filters [Eqs. (14), (17), and (19)] in comparison to the images filtered by the state-of-the art BM3D filter.[14]

Image	σ	MDF [Eqs. (14) and (19)]	Wiener MDF [Eqs. (17) and (19)]	BM3D
Lena	2	43.407	43.546	**43.594**
	5	38.555	38.558	**38.724**
	10	35.488	35.566	**35.932**
	15	33.639	33.795	**34.269**
	20	32.283	32.508	**33.051**
	25	31.211	31.497	**32.071**
	30	30.33	30.668	**31.27**
	35	29.576	29.963	**30.557**
Boats	2	43.101	**43.184**	43.181
	5	37.115	37.181	**37.283**
	10	33.541	33.613	**33.92**
	15	31.576	31.719	**32.14**
	20	30.195	30.388	**30.882**
	25	29.135	29.361	**29.909**
	30	28.282	28.534	**29.117**
	35	27.571	27.844	**28.431**
F-16	2	44.267	44.347	**44.619**
	5	39.016	39.091	**39.527**
	10	35.37	35.524	**36.112**
	15	33.257	33.472	**34.12**
	20	31.765	32.033	**32.711**
	25	30.616	30.933	**31.637**
	30	29.668	30.038	**30.76**
	35	28.857	29.281	**29.985**
Man	2	43.357	43.4	**43.605**
	5	37.34	37.443	**37.816**
	10	33.346	33.503	**33.981**
	15	31.261	31.426	**31.929**

Table 3 (*Continued*).

Image	σ	MDF [Eqs. (14) and (19)]	Wiener MDF [Eqs. (17) and (19)]	BM3D
Man	20	29.896	30.067	**30.589**
	25	28.896	29.082	**29.616**
	30	28.115	28.323	**28.86**
	35	27.471	27.707	**28.224**
Stream & bridge	2	42.553	42.573	**42.662**
	5	35.511	35.605	**35.775**
	10	30.794	30.976	**31.174**
	15	28.44	28.615	**28.789**
	20	26.978	27.126	**27.271**
	25	25.96	26.086	**26.228**
	30	25.195	25.31	**25.46**
	35	24.595	24.703	**24.862**
Aerial	2	43.123	43.08	**43.465**
	5	36.458	36.504	**37.008**
	10	31.992	32.112	**32.521**
	15	29.62	29.777	**30.058**
	20	28.039	28.224	**28.405**
	25	26.867	27.074	**27.181**
	30	25.946	26.167	**26.211**
	35	25.192	**25.423**	25.326
Baboon	2	42.368	**42.406**	42.303
	5	35.173	**35.273**	35.104
	10	30.43	**30.568**	30.394
	15	27.962	**28.135**	27.902
	20	26.351	**26.541**	26.277
	25	25.186	**25.378**	25.115
	30	24.3	**24.482**	24.226
	35	23.597	**23.766**	23.391

Table 3 (*Continued*).

Image	σ	MDF [Eqs. (14) and (19)]	Wiener MDF [Eqs. (17) and (19)]	BM3D
Sailboat	2	42.935	**42.99**	42.839
	5	36.4	**36.555**	36.375
	10	32.628	32.687	**32.708**
	15	30.759	30.844	**30.86**
	20	29.47	**29.604**	29.571
	25	28.465	**28.647**	28.569
	30	27.639	**27.864**	27.737
	35	26.942	**27.203**	26.928
Elaine	2	42.927	**42.974**	42.726
	5	36.678	**36.895**	36.372
	10	33.464	**33.425**	33.352
	15	32.131	32.061	**32.143**
	20	31.276	31.274	**31.296**
	25	30.591	**30.676**	30.585
	30	29.997	**30.168**	29.949
	35	29.457	**29.712**	29.337
Couple	2	43.462	**43.531**	42.939
	5	37.974	**38.032**	37.325
	10	34.223	**34.352**	33.794
	15	32.086	**32.268**	31.759
	20	30.595	**30.823**	30.322
	25	29.444	**29.719**	29.188
	30	28.512	**28.827**	28.244
	35	27.737	**28.079**	27.42
Tiffany	2	43.643	**43.737**	43.669
	5	38.567	38.658	**38.854**
	10	35.244	35.377	**35.671**
	15	33.451	33.601	**33.846**
	20	32.233	32.419	**32.535**
	25	31.296	**31.542**	31.524

Table 3 (*Continued*).

Image	σ	MDF [Eqs. (14) and (19)]	Wiener MDF [Eqs. (17) and (19)]	BM3D
Tiffany	30	30.523	**30.84**	30.653
	35	29.86	**30.246**	29.903
Peppers	2	43.044	**43.19**	42.917
	5	37.497	**37.551**	37.535
	10	34.653	34.636	**34.947**
	15	33.125	33.186	**33.502**
	20	31.985	32.128	**32.371**
	25	31.045	31.265	**31.419**
	30	30.239	30.531	**30.576**
	35	29.531	**29.89**	**29.795**

less (about 9 dB and more increase). Thus, it seems possible to predict $\mathrm{MSE}_{\mathrm{out}}/\sigma^2$ (or, equivalently, $\mathrm{MSE}_{\mathrm{out}}$ for *a priori* known σ^2) from analysis of $P_{2.7\sigma}$ with practically acceptable degree of accuracy. This can be one possible direction of future research. It can be also expected that the use of polynomial threshold operators and other more sophisticated

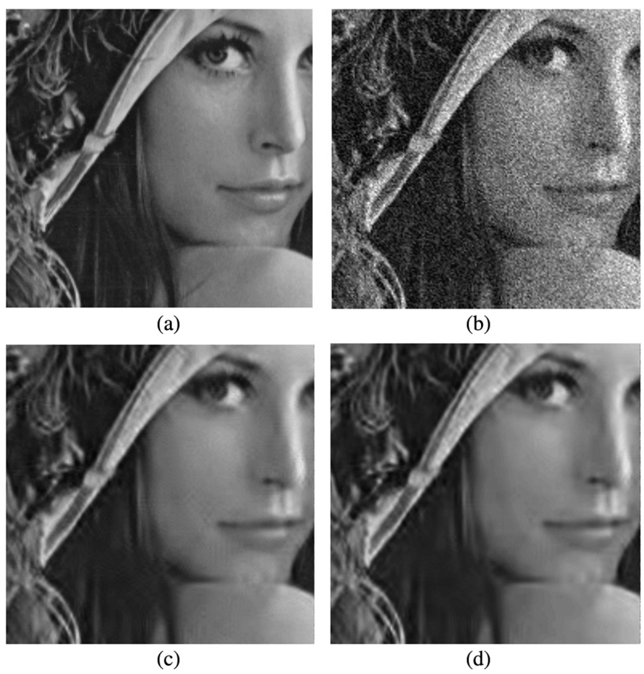


Fig. 3 Filtering results for the test image "Lena" contaminated by AWGN with $\sigma = 25$: (a) a fragment of the original image; (b) a noisy fragment; (c) the proposed MDF filter [Eqs. (14) and (19)] output; and (d) the proposed Wiener MDF [Eqs. (17) and (19)] output. Some blocking effects can be noted on Lena's face in (c).

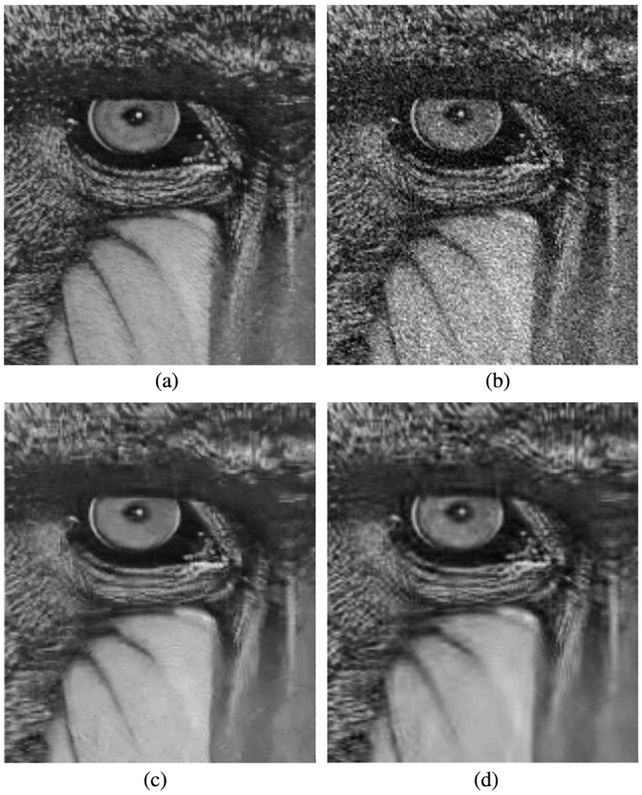


Fig. 4 Filtering results for the test image "Baboon" contaminated by AWGN with $\sigma = 25$: (a) a fragment of the original image; (b) a noisy fragment; (c) the output of the BM3D filter; and (d) the proposed Wiener MDF [Eqs. (17) and (19)] output. The picture in (d) looks more natural.

Table 4 DCT-based filter efficiency and DCT coefficient statistics for different test images and noise variances.

Image	σ	β_{opt}	$P_{2\sigma}$	$P_{2.7\sigma}$	MSE_{out}/σ^2
Baboon	5	2.3	0.340	0.233	0.78
Stream & bridge	5	2.38	0.369	0.204	0.71
Baboon	10	2.34	0.450	0.128	0.58
Man	5	2.45	0.474	0.111	0.46
Stream & bridge	10	2.37	0.474	0.105	0.52
Boats	5	2.38	0.476	0.107	0.49
Baboon	15	2.37	0.501	0.083	0.47
Peppers	5	2.35	0.509	0.076	0.45
F-16	5	2.56	0.518	0.077	0.32
Lena	5	2.5	0.519	0.073	0.36
Stream & bridge	15	2.37	0.521	0.067	0.4
Tiffany	5	2.49	0.523	0.069	0.36
Man	10	2.51	0.536	0.059	0.29
Boats	10	2.56	0.538	0.058	0.29
F-16	10	2.69	0.557	0.046	0.19
Peppers	10	2.63	0.560	0.041	0.22
Man	15	2.57	0.561	0.041	0.21
Lena	10	2.7	0.561	0.042	0.19
Boats	15	2.61	0.561	0.042	0.2
Tiffany	10	2.6	0.566	0.037	0.2
F-16	15	2.74	0.572	0.035	0.14
Lena	15	2.8	0.575	0.032	0.13
Peppers	15	2.77	0.576	0.031	0.14
Tiffany	15	2.7	0.580	0.027	0.13

thresholds[35,36] can improve performance of the DCT-based denoising.

6 Conclusions

Different approaches to filtering grayscale images corrupted by AWGN are considered including the DCT-based denoising with hard thresholding, two-stage Wiener filter, and ideal Wiener filters that are compared to the state-of-the art BM3D technique. Several sizes of fully overlapped image blocks are studied and it is shown that processing in 8×8 and 16×16 pixel blocks produces approximately the same results. It has been demonstrated that the performance can be slightly improved by combining the filter outputs that perform processing using different block sizes. Following this approach, two multiscale DCT-based filters, MDF and Wiener MDF, are proposed and their properties analyzed.

Potential limits of output PSNR (or MSE) for the ideal Wiener filter and Chatterjee's approach are obtained and compared. These limits are, on average, of the same order but can differ by up to 5 dB depending on the image processed and noise variance. Thus, we can state that the potential limits of filtering efficiency are "approach-dependent."

The state-of-the-art filters including the DCT-based denoising and the Wiener-based techniques provide filtering performances quite close to Chatterjee's limit for complex-structure images and large noise variance. Performance characteristics of the state-of-the art BM3D filter and the proposed Wiener MDF are very close while the latter filter is simpler and faster.

The proposed MDF techniques require less computational time than the BM3D filter and, especially, the Chatterjee filter, which requires image clustering to perform nonlocal averaging. MDF technique [Eqs. (14) and (19)] is about two times faster than the Wiener MDF [Eqs. (17) and (19)] and produces good visual quality of the filtered images when the noise variance is low ($\sigma < 0.1$).

It has also been shown that filtering efficiency depends considerably on DCT coefficient statistics. A more detailed study of this dependence can be a direction of future research to further improve performance of the block-wise DCT-based filters.

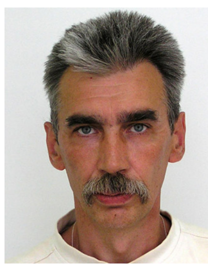
Acknowledgments

We are thankful to anonymous reviewers for their valuable comments and propositions. This work was partially supported by Instituto Politecnico Nacional as a part of the research project SIP20120530.

References

1. W. K. Pratt, *Digital Image Processing*, 4th ed., Wiley-Interscience, New York (2007).
2. A. Bovik, *Handbook of Image and Video Processing*, p. 1429, Academic Press, USA (2000).
3. R. Touzi, "A review of speckle filtering in the context of estimation theory," *IEEE Trans. Geosci. Remote Sens.* **40**(11), 2392–2404 (2002).
4. A. Foi, "Pointwise shape-adaptive DCT image filtering and signal-dependent noise estimation," Ph.D. Thesis, p. 194, Tampere University of Technology, Tampere, Finland (2007).
5. K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*, Springer-Verlag, New York, p. 355 (2000).
6. V. Lukin et al., "Adaptive DCT-based filtering of images corrupted by spatially correlated noise," *Proc. SPIE* **6812**, 68120W (2008).
7. P. Chatterjee and P. Milanfar, "Is denoising dead?" *IEEE Trans. Image Process.* **19**(4), 895–911 (2010).
8. X. Zhu and P. Milanfar, "Automatic parameter selection for denoising algorithms using a no-reference measure of image content," *IEEE Trans. Image Process.* **19**(12), 3116–3132 (2010).
9. V. Lukin et al., "Image filtering: potential efficiency and current problems," in *International Conf. on Acoustics, Speech and Signal Processing (ICASSP 2011)*, Prague, Czech Republic, p. 4, IEEE (2011).
10. D. Fevralev et al., "Efficiency analysis of color image filtering," *EURASIP J. Adv. Signal Process.* **2011**(1), 1–19 (2011).
11. M. Elad, *Sparse and Redundant Representations. From Theory to Applications in Signal and Image Processing*, p. 376, Springer Science+Business Media, Berlin (2010).
12. C. Kervrann and J. Boulanger, "Local adaptivity to variable smoothness for exemplar-based image regularization and representation," *Int. J. Comput. Vision* **79**(1), 45–69 (2008).
13. K. Dabov et al., "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," in *Proc. IEEE Int. Conf. on Image Process. (ICIP 2007)*, San Antonio, Texas, pp. 313–316, IEEE (2007).
14. K. Dabov et al., "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007).
15. S. Haykin, *Adaptive Filter Theory*, 4th ed., Prentice Hall Int. Inc., New Jersey (2002).
16. F. Jin et al., "Adaptive Wiener filtering of noisy images and image sequences," in *IEEE Int. Conf. Image Process. III (ICIP 2003)*, Barcelona, Spain, pp. 349–352, IEEE (2003).
17. J. S. Lee, "Digital image enhancement and noise filtering by use of local statistics," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-2**(2), 165–168 (1980).
18. D. T. Kuan et al., "Adaptive noise smoothing filter for images with signal dependent noise," *IEEE Trans. Pattern Anal Mach. Intell.* **PAMI-7**(2), 165–177 (1985).
19. G. M. Rajathi and R. Rangarajan, "Efficient adaptive Wiener filter with thresholding for better image enhancement," *Eur. J. Sci. Res.* **69**(1), 143–153 (2012).
20. P.-L. Shui and Y.-B. Zhao, "Image denoising algorithm using doubly local Wiener filtering with block-adaptive windows in wavelet domain," *J. Signal Process.* **87**(7), 1721–1734 (2007).
21. S. Suhaila and T. Shimamura, "Image restoration based on edgemap and Wiener filter for preserving fine details and edges," *Int. J. Circuits, Syst. Signal Process.* **5**(6), 618–626 (2011).
22. S. Rahardja and B. Falkowski, "Comparative study of discrete orthogonal transforms in adaptive signal processing," *IEICE Trans. Fundam.* **E-85-A**(8), 1386–1390 (1999).
23. V. V. Lukin et al., "Image filtering based on discrete cosine transform," *Telecommun. Radio Eng.* **66**(18), 1685–1701 (2007).
24. P. Chatterjee and P. Milanfar, "Patch-based near-optimal image denoising," *IEEE Trans. Image Process.* **21**(4), 1635–1649 (2012).
25. A. Kaarna, "Compression of spectral images," in *Vision Systems: Segmentation and Pattern Recognition*, G. Ohinata and A. Dutta, Eds., pp. 269–298, I-Tech, Vienna, Austria (2007).
26. R. Oktem et al., "Locally adaptive DCT filtering for signal-dependent noise removal," *EURASIP J. Adv. Signal Process.* **2007**, 042472 (2007).
27. V. Lukin et al., "Adaptive DCT-based filtering of images corrupted by spatially correlated noise," *Proc. SPIE* **6812**, 68120W (2008).
28. N. N. Ponomarenko et al., "3D DCT based filtering of color and multichannel images," *Telecommun. Radio Eng.* **67**(15), 1369–1392 (2008).
29. V. Lukin et al., "Discrete cosine transform-based local adaptive filtering of images corrupted by nonstationary noise," *J. Electron. Imaging* **19**(2), 15 (2010).
30. K. Egiazarian et al., "Adaptive denoising and lossy compression of images in transform domain," *J. Electron. Imaging* **8**(3), 233–245 (1999).
31. R. Oktem et al., "Transform based denoising algorithms: comparative study," *J. Electron. Imaging* **11**(2), 149–156 (2002).
32. A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed., Prentice Hall, New Jersey (2009).
33. "The USC-SIPI image database, volume 3: miscellaneous," http://sipi.usc.edu/database/database.php?volume=misc (19 September 2012).
34. P. Chatterjee and P. Milanfar, "Fundamental limits of image denoising," 09 February 2012, http://users.soe.ucsc.edu/~priyam/bounds.
35. C. Smith, S. Agaian, and D. Akopian, "A wavelet-denoising approach using polynomial threshold operators," *IEEE Signal Process. Lett.* **15**, 906–909 (2008).
36. V. Lukin, N. Ponomarenko, and K. Egiazarian, "HVS-metric-based performance analysis of image denoising algorithms," in *European Workshop on Visual Information Processing (EUVIP)*, Paris, France, pp. 156–161, IEEE (2011).

Oleksiy Pogrebnyak received his PhD degree from Kharkov Aviation Institute (now National Aerospace University), Ukraine, in 1991. Currently, he is with The Center for Computing Research of National Polytechnic Institute, Mexico. His research interests include digital signal/image filtering and compression, and remote sensing.



Vladimir V. Lukin graduated from Kharkov Aviation Institute (now National Aerospace University) in 1983 and got his diploma with honors in radio engineering. Since then he has been with the Department of Transmitters, Receivers and Signal Processing of National Aerospace University. He defended the thesis of Candidate of Technical Science in 1988 and Doctor of Technical Science in 2002 in DSP for Remote Sensing. Since 1995 he has been in cooperation with Tampere University of Technology. Currently, he is department vice chairman and professor. His research interests include digital signal/image processing, remote sensing data processing, image filtering, and compression.