electronics

MDPI

Article

Adaptive Block-Based Approach to Image Noise Level Estimation in the SVD Domain

Emir Turajlic

Department of Telecommunications, Faculty of Electrical Engineering, University of Sarajevo, 71000 Sarajevo, Bosnia and Herzegovina; emir.turajlic@etf.unsa.ba; Tel.: +387-60-345-6765

Received: 20 October 2018; Accepted: 3 December 2018; Published: 6 December 2018

Abstract: Estimation of additive white Gaussian noise levels in images has a variety of image processing applications including image enhancement, segmentation and feature extraction. Designing an algorithm with a consistent performance across a range of noise levels and image contents is a challenging problem; without any prior information, it is difficult to differentiate the noise signal from the underlying image signal. In this paper, an adaptive block-based noise level estimation algorithm in the singular value decomposition domain is proposed. The algorithm has the ability to change the singular value tail length according to the observed noise levels. A number of different choices of block size are considered and, for each choice, a mathematical model is proposed to describe how to adjust the singular value tail length as a function of the initial noise level estimates. In comparison with a seminal fixed singular value tail length algorithm, the proposed algorithm significantly improves the noise level estimation accuracy at low noise levels at the expense of a small increase in computational time; for example, for the block size of 64 × 64 and AWGN level $\sigma = 1$, the MSE is reduced by 65%, whilst the computational time is increased by less than 1.3%.

Keywords: Additive white Gaussian noise; noise level estimation; block-based estimation; singular value decomposition

1. Introduction

In general, images can be corrupted by a noise signal during image acquisition, transmission, storage and image processing [1,2]. Noise level estimation from a single noisy image without any prior information is an important and challenging field of digital image processing. Noise level estimation algorithms have a variety of image processing applications including image enhancement and denoising [2–5], edge detection [6], image segmentation [7,8], and feature extraction [9,10]. The objective of noise level estimation research is to design an algorithm that can provide accurate and consistent performance across a range of noise levels, image contents, lighting variations, different textures, and contrast levels. In practice, noise level estimation is a very challenging task as it is difficult to ascertain the extent to which the observed local variations are associated with the noise.

Over the past decades of research, various approaches to noise level estimation have been proposed. In the broadest classification, the noise level estimation algorithms can be grouped into one of the following classes: block-based methods, filter-based methods and transform domain methods. In the block-based noise level estimation approach, images are tessellated into blocks [11–15]. Most commonly, this approach entails two additional steps: identification of homogeneous image patches and a statistical analysis of selected image patches to generate a noise level estimate for an entire image. There exists an underlying assumption that the principal signal variations in homogenous image patches are due to noise and not due to the underlying image signal. Performing statistical analysis on homogenous image patches alleviates the influence of image textures on noise level estimation. Filter-based methods use the difference between the noisy image and the low-pass filtered image to estimate the levels of noise in still images [16–18]. It is assumed that low-pass filtering is able to sufficiently suppress the underlying image signal and that the resulting difference-image contains mostly image noise. This assumption does not hold true when the underling image contains more complex structures. Various solutions have been proposed to improve the filter-based noise level estimation in such instances. For example, in [18] adaptive edge detection is employed to exclude edges from the noise level estimation process. Since the filter-based noise level estimation algorithms are usually associated with the higher levels of accuracy and robustness, whilst the block-based algorithms are associated with simplicity and a lower computational load, many hybrid algorithms have been proposed in an attempt to combine the benefits of both approaches [19–21]. Transform-based methods transform the image from a spatial domain into another domain prior to noise level estimation [22–25]. Wavelet transform [23] and DCT transform [25] are popular choices of transform for noise level estimation. Recently, noise level estimation algorithms based on principal component analysis [26,27] and singular value decomposition [28,29] have become increasingly popular.

In a seminal paper [28], the noise level estimation in the Singular Value Decomposition (SVD) domain is demonstrated to attain levels of accuracy and consistency over different image contents that are comparable to the state-of-the-art methods. The algorithm in [28] performs the singular value decomposition on an entire image and, thus, it can be computationally expensive. In [29], an attempt is made to reduce the computational complexity of the noise level estimation in the SVD domain by adopting a block-based approach to noise level estimation.

In this paper, we demonstrate that the performance of both algorithms, [28] and [29], tends to deteriorate at low noise levels. In [28] and [29], the noise level estimates are obtained by analyzing the tail of a singular value sequence. It is assumed that noise has the most significant effect on the tail of the singular value sequence, whereas the underlying image signal mostly contributes to the first few singular values. Both algorithms use the fixed singular value tail length to generate noise level estimates. However, at low noise levels, the effect of noise on the singular value sequence is reduced to a smaller number of singular values. Thus, at low noise levels, the length of singular value sequence that is used in [28] and [29] is not effective and the accuracy of noise level estimation deteriorates.

In this paper, an adaptive block-based algorithm for noise level estimation in the SVD domain is proposed. The proposed algorithm is designed to address the identified deficiency of the algorithm in [29]. Thus, the principal objective of the proposed method is to improve the performance of the block-based noise level estimation algorithm [29] at low noise levels without a significant increase in computational time.

The main contribution of this paper is the proposed block-based adaptive noise level estimation algorithm in the SVD domain that can adapt the length of a singular value tail based on the observed noise levels. The ability to adapt the noise level estimation characteristics according to the observed noise levels is imposed through a two-stage process, whereby initial noise level estimates are used to adjust the algorithm parameters in order to facilitate a more accurate noise level estimation performance in the second stage of the algorithm. A general function for adjusting the singular value tail length with respect to the initial noise level estimates is proposed. The proposed algorithm is implemented for a number of different choices of block size. For each choice of block size, the coefficients of the proposed function are obtained empirically. Furthermore, the optimal choice of block size is evaluate. Specific measures are taken to ensure that the added computational time cost compared to the block-based noise level estimation algorithm in [29] is minimized.

In a comparative analysis with the image-based noise level estimation algorithm in the SVD domain [28] and the block-based noise level estimation algorithm in the SVD domain [29], the proposed noise level estimation algorithm is shown to significantly reduce the noise level estimation mean square error at a negligible computational time cost. The performance at low noise levels is especially improved.

This paper is organized as follows. Noise level estimation in SVD domain is discussed in Section 2. The proposed adaptive block-based noise level estimation algorithm in the SVD domain is presented in Section 3. Simulation results are presented and discussed in Section 4. Section 5 concludes the paper.

2. Noise Level Estimation in the SVD Domain

Singular value decomposition of some real rectangular $m \times n$ matrix A of rank r is defined as:

$$A = USV^T. \tag{1}$$

Here, $U \in R^{mxm}$ and $V \in R^{nxn}$ are orthogonal matrices such that columns of matrix U correspond to eigenvectors of AA^T, whilst the rows of V^T correspond to eigenvectors of A^TA. Matrix S is a diagonal matrix of singular values. Singular values $s(i)$ form a decreasing sequence $s(1) \geq s(2) \ldots \geq s(r) > 0$, where r denotes the matrix rank.

In general, the SVD domain enables separation of the underlying image signal and noise, lending itself particularly well to noise level estimation. Since the paper considers a block-based approach to noise level estimation, the effect of noise on the singular value curve is demonstrated on an image patch rather than on a whole image. Figure 1a presents a test image, and Figure 1b presents an example of a 64 × 64 image patch. In Figure 1c, the singular value curves at various levels of additive white Gaussian noise are presented. If an image is corrupted by an additive noise, the singular values of the resulting image correspond to the sum of singular values due to the signal and noise, $s(i) = s_s(i) + s_n(i)$. In general, noise signal mostly affects the singular value sequence tail, whereas the underlying image signal strongly affects the first few singular values. With the decreasing noise levels, the effect of noise on the singular value curve is reduced to an increasingly smaller number of singular values.

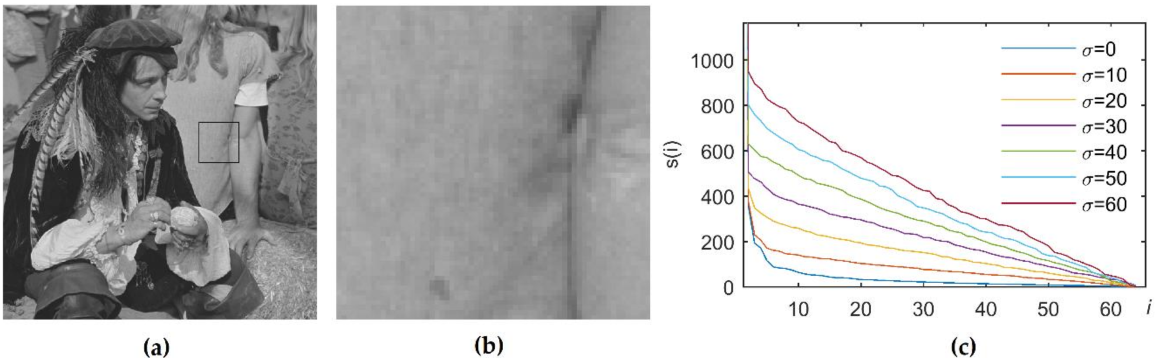


Figure 1. (**a**) Test image "Pirate". An example image block of size 64 × 64 is outlined; (**b**) The example image block is magnified; (**c**) Singular value curves corresponding to the example image block at various noise levels.

The proposed noise level estimation algorithm is based on the approach presented in [28,29]. In a seminal paper on the noise level estimation in the SVD domain [28], for each image, noise level estimation requires that the singular value decomposition is performed on an input image corrupted by an unknown noise source and on an image that is obtained when the input image is further degrade by a known noise source. Multiple SVD decompositions based on the use of a known noise source enables a consistent noise level estimation performance across different image contents. Noise level estimation in the SVD domain entails six distinct steps [28].

In the first step, singular value decomposition is performed on an input image A that is corrupted by an unknown noise source σ resulting in a sequence of singular values, $s^{(A)}(i)$. Subsequently, an average of M trailing singular values is evaluated as:

$$P^{(A)}(M) = \frac{1}{M} \sum\nolimits_{i=r-M+1}^{r} s^{(A)}(i), \tag{2}$$

where $P^{(A)}$ is a function of the singular value tail length M. In [28], the recommended singular value tail length is defined as $M = 0.75r$. Furthermore, in [28], it is ascertained that a reliable noise level estimation requires the length of singular value tail to be restricted to a range $M \in [0.25r, 0.8r]$.

In the next step, the input image A is degraded by a zero mean additive white Gaussian noise with a known deviation $\sigma_B = 50$ to obtain a new image B. This image is processed in a similar fashion as the input image A. First, the singular value decomposition is performed on the image B to obtain a sequence of singular values $s^{(B)}(i)$ and subsequently, the average of M trailing singular values is evaluated:

$$P^{(B)}(M) = \frac{1}{M}\sum\nolimits_{i=r-M+1}^{r} s^{(B)}(i), \tag{3}$$

In the final step, the noise level estimate $\hat{\sigma}$ is obtained as in:

$$\hat{\sigma} = \frac{\alpha\sigma_B^2}{2(P^B(M) - P^A(M))} - \frac{P^B(M) - P^A(M)}{2\alpha}, \tag{4}$$

where the parameter α, defines the slope of a linear function that describes the distribution of average $P(M)$ values across a range of noise levels [28]. Equation (4) is derived in [28] and it is optimal for a given singular value tail length. The value of parameter α depends on the matrix rank r and the singular value tail length M. For each r and M, parameter α is evaluated experimentally, as described in Section 3.2. Figure 2 displays the sequence of steps involved in the noise level estimation in the SVD domain.

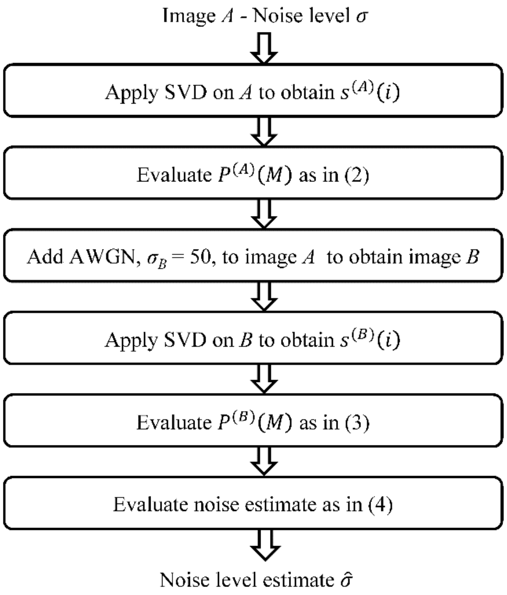


Image A - Noise level σ
Apply SVD on A to obtain $s^{(A)}(i)$
Evaluate $P^{(A)}(M)$ as in (2)
Add AWGN, $\sigma_B = 50$, to image A to obtain image B
Apply SVD on B to obtain $s^{(B)}(i)$
Evaluate $P^{(B)}(M)$ as in (3)
Evaluate noise estimate as in (4)
Noise level estimate $\hat{\sigma}$

Figure 2. Noise level estimation in the SVD domain.

3. Proposed Adaptive Block-Based Noise Level Estimation Algorithm in the SVD Domain

In this section, we present our proposed adaptive block-based algorithm for noise level estimation in the SVD domain. The principal idea of our proposed algorithm is to perform the block-based noise level estimation in the SVD domain in two distinct stages in order to enable the parameters of the algorithm to adapt to the observed noise levels. Specifically, the primary objective is to reduce the length of the singular value tail at low noise levels.

3.1. Overview of the Proposed Noise Level Estimation Algorithm in the SVD Domain

In the first stage of the proposed block-based noise level estimation algorithm, a coarse noise level estimate is generated. The first stage of the algorithm is based on the block-based noise level estimation algorithm in [29]. In the second stage, the algorithm parameters, including the length of the singular value tail, are adjusted according to the initial noise level estimate, and the noise level estimation is performed again. The proposed algorithm reduces the length of a singular value tail at low noise levels. This results in a more accurate noise level estimation. The proposed mechanism is based on the observation that at low noise levels, the underlying image signal dominates the noise in the SVD domain and the effective length of the singular value tail decreases. In each stage of the algorithm, the actual noise level estimation is based on the block-based noise level estimation algorithm in [29].

The algorithm in [29] is a block-based variant of the previously described algorithm in [28], where SVD analysis is performed on the entire image. What follows are the main characteristics of the noise level estimation algorithm in [29]. A noisy input image is tessellated into $r \times r$ blocks. Approximately 40% of blocks are randomly selected. From each block, a single noise level estimate is obtained following the steps presented in Figure 2. Finally, the noise level estimate for the entire image is obtained as the average value of individual block-based noise level estimates. In the design of the proposed algorithm, specific measures are taken to ensure that the added computational load due to a two-stage noise level estimation process is minimized.

The proposed adaptive block-based noise level estimation algorithm in the SVD domain entails the following steps:

(1) Tessellate an input image into $r \times r$ blocks.

(2) Randomly select 40% of blocks and perform the singular value decomposition on each block to obtain a sequence of singular values $s_j(i)$, where $i = 1, 2, \dots\ r$, and j denotes the block index, where $j = 1, \dots\ D$. Here, D represents a total number of selected blocks.

(3) Each image block is processed via a sequence of steps presented in Figure 2. For each jth block of image A, the average value of the singular value tail is evaluated using (2) to obtain $P_j^{(A)}(M)$. The singular value tail length is set to $M = 0.75r$. An AWGN with deviation $\sigma_B = 50$ is added to the selected image patches to obtain a new set of image patches that are now associated with the image B. For each jth block of image B, the average value of the singular value tail is evaluated using (3) to obtain $P_j^{(B)}(M)$. From each jth block, a noise level estimate $\hat{\sigma}_j$ is obtained using (4). The initial noise level estimate for the entire image is obtained as the average value of D independent noise level estimates. Each of these D noise level estimates is associated with one image block. These estimates are obtained using the initialized singular value length M and parameter α values. This denotes the end of the first stage of the proposed noise level estimation algorithm.

(4) If the coarse noise level estimate, obtained in the first stage of the proposed noise level estimation algorithm, is below a specified threshold for a give block size, the algorithm enters the second stage. Otherwise, the algorithm is terminated, and the final noise level estimate is equated to the initial noise level estimate.

(5) The second stage of the proposed algorithm adopts the following results from the first stage: $s_j^{(A)}(i)$, $s_j^{(B)}(i)$ and $\hat{\sigma}$.

(6) The length of the singular value tail is adjusted according to the initial noise level estimate. A simple function is proposed to define the relationship between the adjusted M' value and the initial noise levels estimate $\hat{\sigma}$. Here, the apostrophe is placed next to the variable to denote that the variable has been adjusted in the second stage of the algorithm.

(7) Parameter α is adjusted based on the modified singular value tail length M'.

(8) $P_j^{(A)}(M')$ and $P_j^{(B)}(M')$ are evaluated using the adjusted singular value tail length M' and Equations (2) and (3), respectively. Furthermore, the singular value sequences $s_j^{(A)}(i)$, $s_j^{(B)}(i)$ are adopted from the first stage and thus, the singular value decomposition is avoided in the second stage of the algorithm.

(9) From each jth block, a noise level estimate $\hat{\sigma}_j'$ is obtain using (4) and the adjusted parameter α'. The final noise level estimate for the entire image is obtained as the average of D independent

noise level estimates, where each estimate is associated with one image block. The individual estimates are based on the adjusted singular value tail length M' and the adjusted parameter α'. This denotes the end of the second stage of the algorithm.

3.2. Determining the Value of the Slope Paramter α

The important aspect of the noise level estimation in the SVD domain is to determine the value of parameter α. Parameter α defines the slope of a linear function that describes the distribution of average $P(M)$ values across a range of noise levels [28]. The value of parameter α depends on the matrix rank r and the singular value tail length M.

For a given block of size $r \times r$ and the singular value tail length M, the procedure of evaluating parameter α is defined as follows. Average $P(M)$ values are evaluated over a range of different noise levels. For each noise level, $P(M)$ is evaluated on 50 different pure AWGN images and an average value is recorded. Linear regression is used to represent the distribution of the average $P(M)$ values across a range of noise levels and the value of parameter α is equated with the line gradient.

In order to further clarify how parameter α is evaluated, three different choices of block size are considered: 32 × 32, 64 × 64 and 128 × 128. The length of singular value tail is fixed to $M = k \times r$, where $k = 0.75$. Here, parameter k represents a fraction of the singular value tail length with respect to the matrix rank, $k = M/r$. Thus, it denotes the normalized singular value tail length. For the considered choices of block size, the distribution of average $P(M)$ values across a range of AWGN levels is shown in Table 1.

Table 1. Average $P(M)$ values across a range of noise levels for different choices of block size.

Block Size	Noise Level, σ															
	10	13	16	19	22	25	28	31	34	37	40	43	46	49	52	55
32 × 32	34.03	44.38	54.64	64.66	75.13	85.68	95.47	105.73	116.09	126.44	136.95	147.06	156.63	166.05	177.33	188.04
64 × 64	48.41	62.75	77.71	92.29	106.69	120.87	135.98	150.27	164.64	179.61	193.63	208.48	222.75	237.31	252.43	266.41
128 × 128	68.76	89.26	110.05	130.81	151.01	171.88	192.20	213.55	233.85	254.18	274.84	295.64	315.90	336.92	357.17	377.57

In Figure 3, a scatter plot of the average $P(M)$ values against the noise levels is shown for each considered choice of block size. The associated results of least-square fitting are also shown. Clearly, in each case, the gradient is positive and its value increases with the block size. For the considered choices of block size and for the fixed length of singular value tail $M = 0.75r$, the estimated α values are as follows: $\alpha_{32} = 3.4093$, $\alpha_{64} = 4.8467$, $\alpha_{128} = 6.8675$.

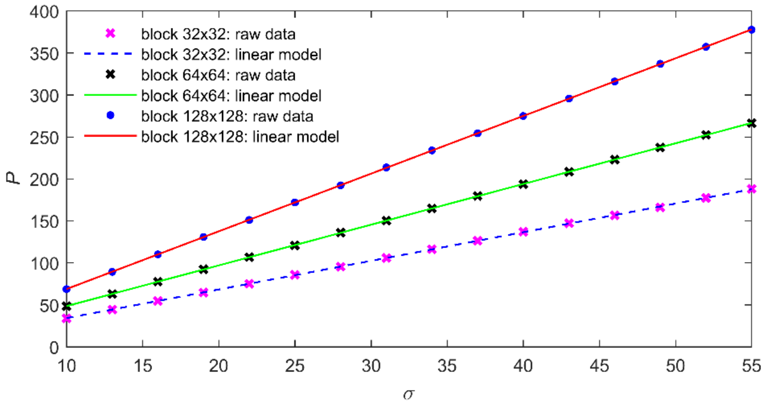


Figure 3. The average $P(M)$ values across a range of noise levels and the associated results of least-square fitting.

The proposed algorithm requires that parameter α is specified for each block size $r \times r$ and the singular value tail length $M = k \times r$. Thus, the aforementioned procedure is repeated for each choice of block size and across a range of singular value tail lengths. In each case, the considered singular value tail length M is restricted to range $M \in [0.25r, 0.8r]$ in order to ensure accurate noise level estimation [28]. For example, for the block size of 128 × 128, the considered singular value tail range is $M \in [32, 102]$.

For each considered block size and the singular value tail length, the associated parameter α is evaluated and the results are presented in Figure 4. The singular value tail lengths are normalized and represented by parameter k.

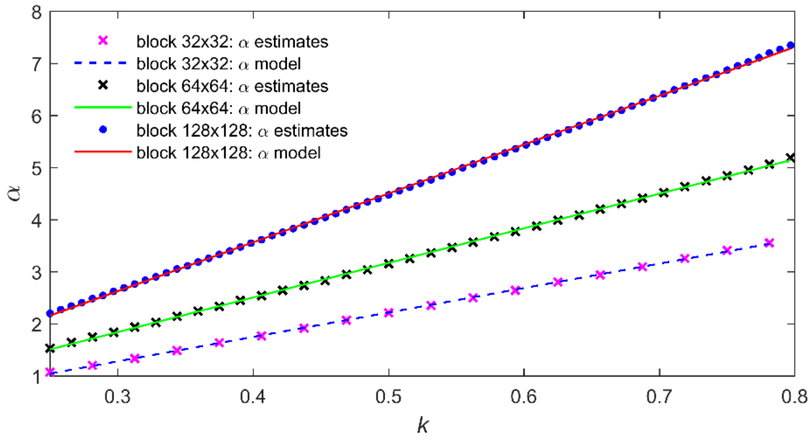


Figure 4. The scatterplot of parameter α against the normalized singular value tail length k and the proposed linear model.

Given the nature of parameter α distribution over the range of the normalized singular value tail lengths, a simple model is proposed to represent the estimate $\hat{\alpha}$ as a linear function with respect to parameter k:

$$\hat{\alpha}(k) = a_1 k + a_0. \tag{5}$$

For each choice of block size, the coefficients of the proposed model, a_0 and a_1 are evaluated using least-squares fitting, and the results are presented in Table 2 and Figure 4. In the implementation of the proposed method, estimated α values can be stored in a lookup table or alternatively, the proposed linear model in (5) can be used.

Table 2. The coefficients of the proposed α model for different choices of image block size.

Block Size	a_1	a_0	MSE
32 × 32	4.6891	−0.1262	1.3936×10^{-4}
64 × 64	6.6399	−0.1501	3.3265×10^{-4}
128 × 128	9.3753	−0.1852	5.8050×10^{-4}

3.3. Adjusting the Singular Value Tail Length Based on the Initial Noise Level Estimates

The main idea behind the proposed algorithm is to enable the length of the singular value tail to be adjusted according to the initial noise level estimates. The adaptive change in the singular value tail length enables a more consistent performance across a range of noise levels.

In this section, a simple model is proposed to represent the normalized singular value tail length as a function of the initial noise level estimate. In this paper, three different choices of block size are considered, 32 × 32, 64 × 64, and 128 × 128, and for each choice of block size, a specific model is developed.

In order to develop a function of the normalized singular value tail length with respect to a noise level estimate, what needs to be examined is how the noise level estimation performance is affected by the normalized singular value tail length at various noise levels. For each block size, the following experiment is conducted. The noise level estimation mean square error is evaluated for range of k

values on a dataset of N = 20 standard images, Supplementary A, Figure S1. The simulation is repeated L = 40 times for each noise level σ and MSE results are reported. The mean square estimation error is evaluated as:

$$E_{MSE} = \frac{1}{L \times N} \sum\nolimits_{n=1}^{N} \sum\nolimits_{l=1}^{L} (\sigma_n - \hat{\sigma}_{n,l})^2, \tag{6}$$

where $\hat{\sigma}$ denotes the noise level estimate.

The normalized singular value tail length is restricted to a range $k \in [0.25, 0.8]$. This restriction is placed to ensure accurate noise level estimation [28]. For the block size of 128 × 128, the singular value tail length is varied from $k = 0.25$ to $k = 0.8$, in increments of $\Delta k = 0.156$. This corresponds to the tail length variation from M = 32 to M = 102, in increments of ΔM = 2. The results are reported in Appendix A, Table A1. Similarly, for the block size of 64 × 64, the normalized singular value tail length is varied from $k = 0.25$ to $k = 0.8$, in increments of $\Delta k = 0.156$, which corresponds to the singular value tail length variation from M = 16 to M = 51, in increments of ΔM = 1. The results are reported in Appendix A, Table A2. For block size of 32 × 32, the normalized singular value tail length is varied from $k = 0.25$ to $k = 0.8$, in increments of $\Delta k = 0.313$, which corresponds to the singular value tail length change from M = 8 to M = 25, in increments of ΔM = 1. The results are reported in Appendix A, Table A3.

For each block size, similar trends can be observed regarding MSE distribution across noise levels. For a given noise level, there is some normalized singular value tail length k that minimizes the estimation MSE level. In general, as parameter k deviates from the optimal value, in either direction, the associated MSE value deteriorates. More importantly, as the noise level increases, the optimum parameter k value tends to increase. These results show that significant improvements in MSE level can be achieved by adapting the singular value tail length according to a specific noise level.

In this paper, we propose a simple function to represent how to adjust the normalized singular value tail length k according to the noise level σ:

$$k' = f(\sigma, b_0, b_1) = \begin{cases} 0.8 & for\ \sigma > 0.8 \\ 0.25 & for\ \sigma < 0.25, \\ 0 & b_1\sigma + b_0 \end{cases} \tag{7}$$

where b_0 and b_1 are coefficients of the proposed model. Here, k' denotes the adjusted normalized singular value tail length. The coefficients of the proposed function are obtained by solving the following optimization problem:

$$\underset{b_0,\, b_1}{\arg\min} \frac{1}{N} \sum\nolimits_{n=1}^{N} (\sigma_n - \hat{\sigma}_{b_0,\, b_1})^2. \tag{8}$$

Here, $\hat{\sigma}_{b_0,\, b_1}$ denotes the estimate of the noise level σ_n that is obtained using the coefficients b_0 and b_1 to describe the singular value tail length adjustment function. Tables A1–A3 show the estimation mean square error results at discrete values of singular value tail length k, and noise level σ_n. Since parameter k is inherently discrete, the output of function f is rounded to the nearest discrete k value. For three different choices of block size, particle swarm optimization [30] is used to solve the optimization problem, and the results are reported in Table 3. The associated functions $f(\sigma, b_0, b_1)$ are presented in Figure 5.

Table 3. The coefficients of the $f(\sigma, b_0, b_1)$ function for three different choices of image block size.

Block Size	b_1	b_0	Min E_{MSE}
32 × 32	0.097059	0.347166	0.446314
64 × 64	0.040105	0.438450	0.399865
128 × 128	0.022484	0.543005	0.461783

The proposed algorithm uses a threshold value to determine whether or not the algorithm enters the second stage, where the noise level estimation is performed based on the adjusted singular value tail length. For each choice of block size, the threshold value σ^{th} is determined as the noise level at which the function describing the adaptation of the singular value tail length according to the noise levels takes the value $k = 0.75$. This normalized singular value tail length $k = 0.75$ is recommended in [28] and [29] as it offers good noise level estimation performance at mid to high noise levels. The decrease in the singular value tail length is only effective at low noise levels. Thus, the noise level at which $k = 0.75$ is used as a threshold value to decide whether or not to adjust the singular value tail length. When the initial noise level is above the given threshold σ^{th}, the proposed algorithm and the noise level estimation algorithm in [29] behave identically. For the three choices of block size, the following threshold values are ascertained $\sigma^{th}_{32} = 4.15$, $\sigma^{th}_{64} = 7.77$, and $\sigma^{th}_{128} = 9.21$.

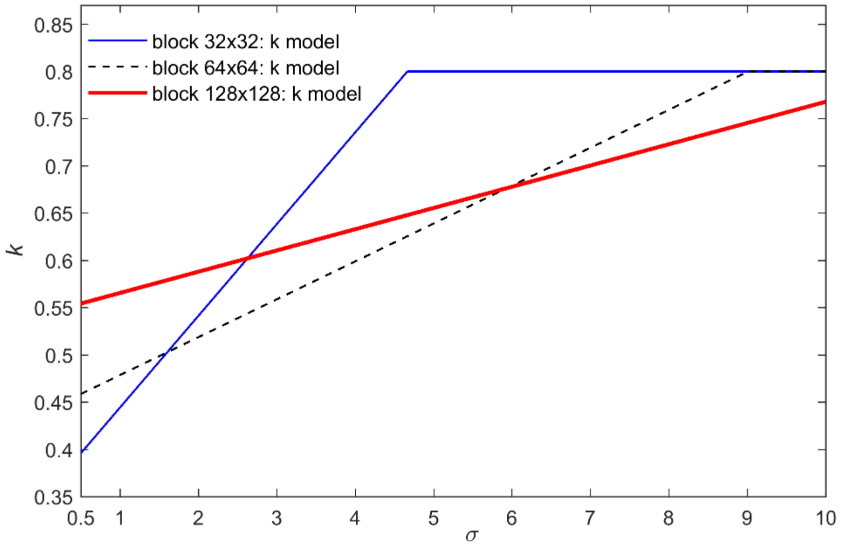


Figure 5. The proposed functions for adjusting the singular value tail length according to the initial noise level estimates for three different choices of block size.

4. Results and Discussion

In this section, a comparative analysis of the proposed block-based noise level estimation algorithm in the SVD domain with the alternative SVD-domain-based algorithms in [28] and [29] is presented.

In order to simplify the discussion and reporting of the experimental results, the image-based algorithm in [28] is referred to as simply SVD. On the other hand, the proposed adaptive algorithm is abbreviated as ASVD, whilst the block-based noise level estimation algorithm in [29] is abbreviated as SVD-B. The number following the abbreviation is used to identify the choice of block size. Thus, the abbreviation ASVD 64 denotes the proposed adaptive block-based noise level estimation algorithm implemented using 64 × 64 image blocks. The abbreviation SVD-B 32 denotes the block-based noise level estimation algorithm in [29] implemented using 32 × 32 image blocks.

In Section 4.1, the proposed ASVD algorithm is compared to SVD-B and SVD noise level estimation algorithms in terms of MSE and average estimation error. The block-based methods are evaluated for three different choices of block size, 32 × 32, 64 × 64, and 128 × 128. In Section 4.2, the proposed noise level estimation is compared to the two alternative SVD-domain-based noise level estimation algorithms, in terms of computational time.

Noise level estimation algorithms are implemented using MATLAB 2017 (The MathWorks, Inc., Natick, MA, USA), and all the experiments are performed on Intel's i7-4510U 2.GHz CPU with 8 GB of RAM. The average noise level estimation error, MSE, and computational time results are used to ascertain the optimal choice of block-size for the proposed adaptive block-based noise level estimation algorithm. In the implementation of the algorithm, MATLAB function "s = svd(A)" is used to perform singular value decomposition of image blocks. In the second stage of the proposed algorithm, the adjusted singular value tail length is obtained using the function (7), where the coefficients b_0 and b_1 are determined for different choices of image block size as in Table 3. The adjusted singular value tail length requires the parameter α to be adjusted as well. In the implementation of the proposed algorithm, a set of parameter α values corresponding to different block size choices and singular value tail lengths is stored in a lookup table.

4.1. MSE and Average Noise Level Estimation Error

The performance of the proposed block-based noise level estimation algorithm in the SVD domain is compared to the SVD-B algorithm on a set of test images shown in Figure 6. Noise level estimation performance is evaluated over a range of different noise levels. Noise levels are varied from $\sigma = 1$ to $\sigma = 10$ in increments of $\Delta\sigma = 0.5$. The implementation of the proposed algorithm uses the lookup table for parameter α.

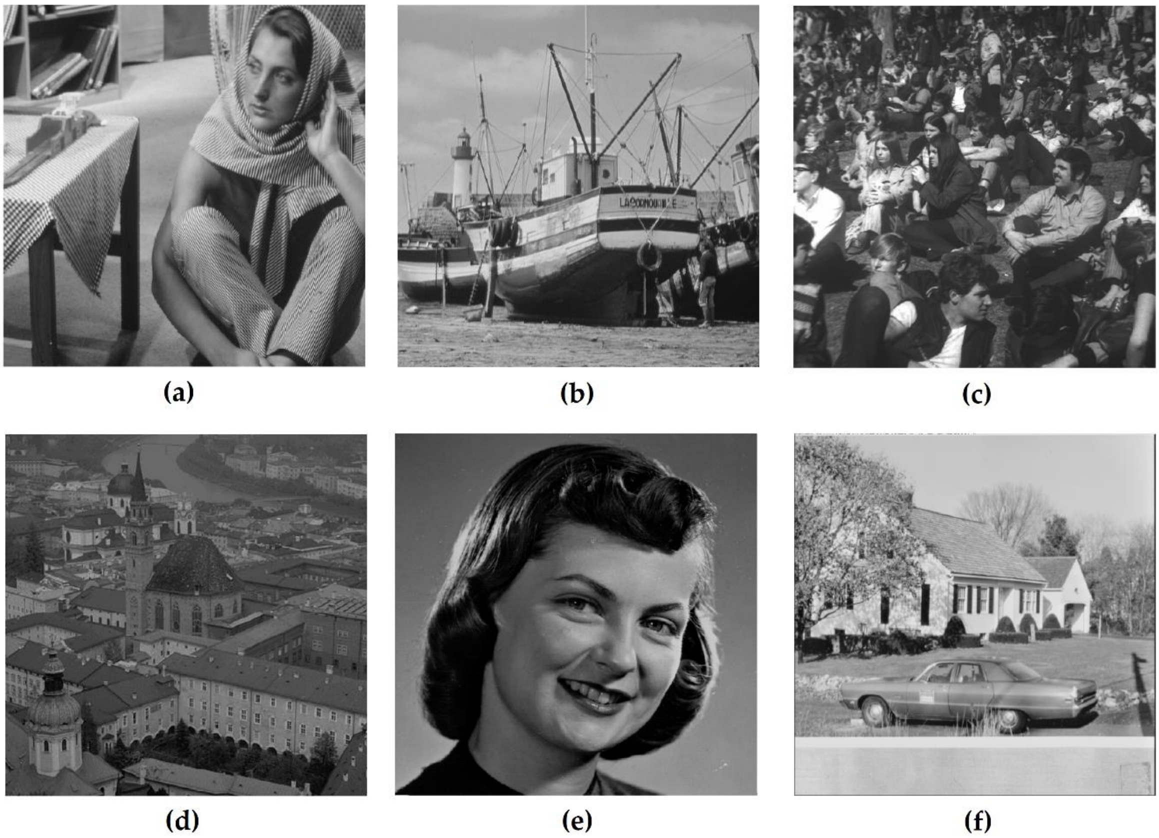


Figure 6. Test images: (**a**) Barbara; (**b**) Boat; (**c**) Crowd; (**d**) Town; (**e**) Girlface; (**f**) Car.

For each noise level, simulations are repeated 50 times and MSE results are presented in Table 4. Both, the proposed ASVD algorithm and the SVD-B [29] algorithm are evaluated for three different choices of block size, 32 × 32, 64 × 64, and 128 × 128.

The results in Table 4 show that SVD-B has a very important limitation. For all considered test images and choices of block size, the estimation MSE levels significantly deteriorate with the decreasing noise levels, as explained below. The proposed ASVD algorithm is designed to solve this problem. Specifically, the ability to adapt the length of singular value tail according to the initial noise level estimates enables the proposed algorithm to perform accurate noise level estimation even at low noise level range.

Table 4. Mean square error (MSE) levels for a range of noise levels and test images.

Test Image	Method	Noise Level, σ																			
		0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10
Barbara	SVD-B 128	4.90	3.44	2.27	1.77	1.16	1.24	0.89	0.39	0.424	0.328	0.346	0.278	0.227	0.116	0.144	0.118	0.022	0.063	0.026	0.095
	ASVD 128	0.74	0.28	0.11	0.11	0.02	0.05	0.04	0.02	0.031	0.017	0.030	0.025	0.043	0.056	0.036	0.080	0.028	0.066	0.026	0.095
	SVD-B 64	3.89	2.75	1.68	1.15	0.76	0.68	0.70	0.38	0.259	0.210	0.140	0.081	0.076	0.097	0.035	0.042	0.053	0.029	0.048	0.036
	ASVD 64	0.28	0.09	0.02	0.01	0.03	0.05	0.02	0.03	0.041	0.029	0.044	0.058	0.020	0.065	0.037	0.042	0.053	0.029	0.048	0.036
	SVD-B 32	3.64	2.41	1.59	1.17	0.86	0.59	0.42	0.33	0.249	0.210	0.137	0.119	0.057	0.073	0.071	0.051	0.029	0.027	0.047	0.039
	ASVD 32	0.68	0.30	0.19	0.20	0.20	0.32	0.40	0.33	0.249	0.210	0.137	0.119	0.057	0.073	0.071	0.051	0.029	0.027	0.047	0.039
Boat	SVD-B 128	2.96	2.06	1.04	0.95	0.53	0.30	0.25	0.18	0.155	0.115	0.066	0.060	0.036	0.010	0.013	0.013	0.036	0.014	0.011	0.009
	ASVD 128	0.50	0.23	0.08	0.03	0.02	0.01	0.01	0.02	0.030	0.008	0.017	0.027	0.017	0.029	0.040	0.018	0.060	0.017	0.011	0.009
	SVD-B 64	2.94	1.69	1.10	0.73	0.67	0.36	0.24	0.21	0.107	0.098	0.070	0.035	0.033	0.020	0.020	0.025	0.029	0.015	0.017	0.020
	ASVD 64	0.33	0.16	0.04	0.02	0.02	0.02	0.01	0.02	0.016	0.006	0.008	0.012	0.028	0.017	0.021	0.025	0.029	0.015	0.017	0.020
	SVD-B 32	2.61	1.74	1.16	0.81	0.57	0.38	0.29	0.19	0.137	0.157	0.074	0.085	0.039	0.042	0.030	0.038	0.029	0.049	0.012	0.027
	ASVD 32	0.52	0.27	0.13	0.11	0.08	0.08	0.26	0.19	0.137	0.157	0.074	0.085	0.039	0.042	0.030	0.038	0.029	0.049	0.012	0.027
Crowd	SVD-B 128	2.31	1.89	1.08	0.85	0.50	0.40	0.30	0.18	0.132	0.055	0.028	0.016	0.021	0.012	0.024	0.012	0.039	0.049	0.044	0.059
	ASVD 128	0.13	0.05	0.02	0.02	0.05	0.06	0.07	0.06	0.064	0.078	0.116	0.122	0.072	0.060	0.046	0.048	0.076	0.067	0.044	0.059
	SVD-B 64	2.05	1.39	1.07	0.75	0.47	0.27	0.15	0.10	0.042	0.041	0.041	0.026	0.013	0.029	0.014	0.024	0.053	0.046	0.023	0.018
	ASVD 64	0.02	0.05	0.04	0.07	0.09	0.09	0.15	0.11	0.110	0.076	0.044	0.043	0.036	0.051	0.024	0.024	0.053	0.046	0.023	0.018
	SVD-B 32	1.90	1.14	0.75	0.45	0.34	0.19	0.12	0.08	0.046	0.054	0.012	0.025	0.017	0.016	0.013	0.014	0.024	0.030	0.028	0.038
	ASVD 32	0.01	0.03	0.02	0.03	0.02	0.02	0.02	0.08	0.046	0.054	0.012	0.025	0.017	0.016	0.013	0.014	0.024	0.030	0.028	0.038
Town	SVD-B 128	6.63	4.90	3.57	2.76	1.97	1.60	1.17	0.96	0.892	0.621	0.607	0.443	0.356	0.346	0.277	0.277	0.235	0.220	0.221	0.238
	ASVD 128	2.92	1.88	1.31	0.88	0.58	0.48	0.34	0.35	0.332	0.230	0.266	0.217	0.188	0.220	0.209	0.231	0.212	0.220	0.221	0.238
	SVD-B 64	5.85	4.19	3.02	2.55	1.80	1.38	1.00	0.83	0.718	0.495	0.429	0.382	0.349	0.267	0.146	0.226	0.218	0.218	0.148	0.123
	ASVD 64	2.01	1.19	0.76	0.59	0.40	0.33	0.24	0.22	0.252	0.189	0.210	0.201	0.280	0.254	0.146	0.226	0.218	0.218	0.148	0.123
	SVD-B 32	5.34	3.72	2.63	1.90	1.55	1.01	0.90	0.72	0.578	0.434	0.370	0.379	0.286	0.280	0.206	0.287	0.201	0.182	0.175	0.199
	ASVD 32	2.57	1.60	1.14	0.90	1.12	0.93	0.90	0.72	0.578	0.434	0.370	0.379	0.286	0.280	0.206	0.287	0.201	0.182	0.175	0.199
Girlface	SVD-B 128	2.07	0.96	0.78	0.54	0.26	0.14	0.13	0.09	0.066	0.023	0.029	0.018	0.014	0.020	0.015	0.016	0.011	0.018	0.021	0.010
	ASVD 128	0.77	0.28	0.16	0.10	0.04	0.01	0.02	0.01	0.004	0.007	0.014	0.017	0.011	0.015	0.021	0.022	0.012	0.022	0.021	0.010
	SVD-B 64	2.58	1.30	0.86	0.60	0.52	0.29	0.25	0.18	0.119	0.106	0.051	0.076	0.050	0.061	0.024	0.020	0.035	0.012	0.022	0.024
	ASVD 64	0.96	0.45	0.23	0.15	0.15	0.07	0.06	0.06	0.035	0.036	0.017	0.045	0.032	0.055	0.024	0.020	0.035	0.012	0.022	0.024
	SVD-B 32	2.52	1.61	1.15	0.88	0.53	0.42	0.32	0.26	0.130	0.184	0.126	0.099	0.084	0.050	0.051	0.041	0.056	0.091	0.089	0.030
	ASVD 32	1.18	0.63	0.43	0.41	0.23	0.28	0.31	0.26	0.130	0.184	0.126	0.099	0.084	0.050	0.051	0.041	0.056	0.091	0.089	0.030
Car	SVD-B 128	2.91	1.48	1.11	1.35	1.32	0.85	0.80	0.50	0.435	0.365	0.131	0.219	0.114	0.249	0.297	0.174	0.036	0.100	2.906	1.484
	ASVD 128	0.61	0.20	0.16	0.13	0.22	0.17	0.15	0.05	0.065	0.091	0.024	0.065	0.066	0.122	0.211	0.141	0.030	0.102	0.611	0.198
	SVD-B 64	2.98	2.59	2.04	1.19	1.12	0.66	0.67	0.38	0.586	0.312	0.345	0.202	0.291	0.202	0.058	0.111	0.175	0.076	2.980	2.586
	ASVD 64	0.33	0.20	0.19	0.07	0.07	0.03	0.06	0.02	0.137	0.049	0.109	0.092	0.220	0.183	0.058	0.111	0.175	0.076	0.329	0.204
	SVD-B 32	3.26	2.45	1.68	1.43	1.12	0.79	0.67	0.62	0.463	0.427	0.327	0.268	0.251	0.280	0.209	0.251	0.228	0.117	3.256	2.454
	ASVD 32	0.61	0.50	0.35	0.38	0.47	0.58	0.67	0.62	0.463	0.427	0.327	0.268	0.251	0.280	0.209	0.251	0.228	0.117	0.608	0.499

For example, on test images ‘Barbara’ and ‘Boat’, and at AWGN level $\sigma = 0.5$, the SVD-B 64 algorithm attains the mean square error levels, MSE = 3.888 and MSE = 2.942, respectively. In contrast, on the same test images, ‘Barbara’ and ‘Boat’, and at same noise level $\sigma = 0.5$, the proposed ASVD 64 algorithm attains the mean square error levels of MSE = 0.283 and MSE = 0.332. Thus, in these two instances, the proposed ASVD 64 algorithm offers the improvements in MSE levels over the SVD-B 64 algorithm by more than 92% and 88%, respectively.

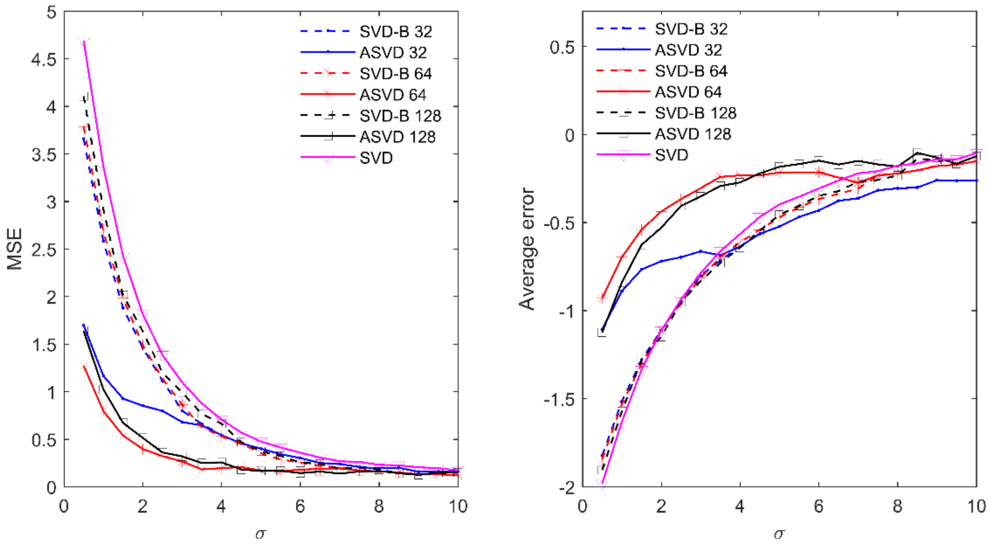
In order to further demonstrate the level of improvement in MSE levels, please consider the following examples. For image “Barbara”, at noise levels $\sigma = \{0.5,\ 2,\ 3.5,\ 5,\ 6.5\}$, the proposed ASVD 64 algorithms offers the following improvements in MSE levels over SVD-B 64 algorithm: 92.8%, 99.1%, 97.1%, 86.2%, 73.7%, respectively. Similarly, for image “Boat”, at noise levels $\sigma = \{0.5,\ 2,\ 3.5,\ 5,\ 6.5\}$, the proposed ASVD 64 algorithms offers the following improvements in MSE levels over SVD-B 64 algorithm: 88.8%, 97.3%, 95.8%, 93.9%, 15.2%, respectively. For image “Town”, at noise levels $\sigma = \{0.5,\ 2,\ 3.5,\ 5,\ 6.5\}$, the proposed ASVD 64 algorithms offers the following improvements in MSE levels over SVD-B 64 algorithm: 65.6%, 76.8%, 76.0%, 61.8%, 19.9%, respectively. In case of image “Girlface”, at noise levels $\sigma = \{0.5,\ 2,\ 3.5,\ 5,\ 6.5\}$, the proposed ASVD 64 algorithms offers the following improvements in MSE levels over SVD-B 64 algorithm: 62.8%, 75.0%, 76.0%, 66.0%, 36.0%, respectively. In case of image “Car”, at noise levels $\sigma = \{0.5,\ 2,\ 3.5,\ 5,\ 6.5\}$, the proposed ASVD 64 algorithms offers the following improvements in MSE levels over SVD-B 64 algorithm: 88.9%, 94.1%, 91,0%, 84.3%, 24.4%, respectively. The improvements in MSE levels usually occur up to the threshold value that is associated with the particular block size. There are examples where that is not the case. Consider the image “Crowd”; the improvements in MSE levels occur up to $\sigma = 3.5$. The result in Table 4 show the proposed ASVD algorithm outperform the SVD-B algorithm for all choices of block size. In each case, the improvements in the noise level estimation performance are significant.

Also, a very general observation is that the performances of the proposed ASVD algorithm and the SVD-B [29] algorithm become more similar with increasing noise levels. At higher noise levels, the initial noise level estimates increase in value and thus, the singular value tail lengths increase as well. As the difference in the singular value tail lengths between the two algorithms gets smaller, the associated noise level estimation performances become more similar. When the initial noise level estimate is above the threshold value, specified for a given block size, the proposed noise level estimation algorithm and the SVD-B algorithm behave identically. In this case, the proposed algorithm will not enter the second stage and the final noise level estimate is equated with the initial noise level estimate.

In the next experiment, the performance of the proposed ASVD algorithm is compared to two SVD-domain-based algorithms, SVD-B [29] and SVD [28]. The two block-based algorithms are implemented for three different choices of block size: 32 × 32, 64 × 64, and 128 × 128. The noise level estimation performance is measured in terms of MSE, as defined in (6), and the average estimation error as in:

$$E_{av} = \frac{1}{L \times N} \sum\nolimits_{n=1}^{N} \sum\nolimits_{l=1}^{L} (\sigma_n - \hat{\sigma}_{n,l}). \tag{9}$$

The algorithms are evaluated over a range of noise levels. Noise levels are varied from $\sigma = 0.5$ to $\sigma = 10$, in increments of $\Delta\sigma = 0.5$. A dataset of N = 15 standard test images is used, and simulations are repeated L = 40 times for each image. Thus, for each noise level, MSE and average estimation error are evaluated on a set of 600 images. The MSE and the average estimation error results are presented in Figure 7.



(a) **(b)**

Figure 7. (**a**) MSE across noise levels; (**b**) Average error across noise levels.

The results show that the proposed ASVD algorithm outperforms the block-based algorithm SVD-B for all choices of block sizes. Both block-based algorithms outperform the algorithm SVD [28], where the noise level estimation is based on the singular value decomposition of the entire images, rather than on image patches. The MSE curves demonstrate that the proposed algorithm offers the best performance for the block size of 64 × 64, whereas the SVD-B algorithm [29] attains the best MSE results for the block size of 128 × 128. Clearly, the choice of block size affects the performance of both algorithms.

Compared to the SVD-B algorithm, the proposed ASVD method significantly improves the MSE results, especially at very low noise levels. For example, for the block size of 64 × 64 and AWGN level $\sigma = 1$, the noise level estimation MSE is reduced by more than 65%. At noise levels $\sigma = 4$ and $\sigma = 6$, the improvements are about 63%, and 29%, respectively. As the level of AWGN gets closer to the threshold value $\sigma_{64}^{th} = 7.77$, the performance of the two algorithms becomes more similar. In instances when the initial noise level estimates are above the $\sigma_{64}^{th} = 7.77$ threshold, the two algorithms have identical noise level estimation performances.

The average error results show that all SVD domain algorithms tend to overestimate the levels of noise at low noise levels. The proposed ASVD algorithm reduces the extent of overestimation. Figure 7 clearly show that the choice of block size has an effect on the performance of both noise level estimation algorithms. However, the proposed algorithm is more affected by the choice of block size compared to the SVD-B [29]. In case of the proposed ASVD algorithm, the choice of block size determines the choice of a function that governs the change of the singular value tail length with respect to the initial noise level estimates. These functions are significantly different for various choices of block size. Furthermore, the range of noise levels where the change in the singular value tail length occurs is a function of the threshold value, which in turn is a function of block size. The block size of 32 × 32 is associated with the smallest threshold value $\sigma_{32}^{th} = 4.15$, while the block size of 128 × 128 is associated with the highest threshold value $\sigma_{128}^{th} = 9.21$.

Figure 7 shows that the proposed ASVD algorithm offers the best performance for the choice of block size 64 × 64. This choice of block size is slightly better than 128 × 128, whilst for the block size of 32 × 32, the performance of the proposed algorithm is considerably worse. In addition, in the case of block size 32 × 32, improvements in MSE levels over the SVD-B algorithm are also restricted to a smaller noise level range. At low noise levels, both block-based approaches outperform the algorithm in [29] in terms of MSE.

4.2. Computational Time

The computational complexity is an important issue in the algorithm design. In Table 5, the average computational times for different SVD-domain-based noise level estimation algorithms are presented. The computational time results correspond to 512 × 512 images.

Table 5. Average computational time for the noise level estimation algorithms in the SVD domain.

Method	SVD	SVD-B 32	ASVD 32	SVD-B 64	ASVD 64	SVD-B 128	ASVD 128
Time (s)	0.0811	0.0590	0.0596	0.0396	0.0401	0.0403	0.0408

All considered SVD block-based noise level estimation algorithms significantly reduce the computational time compared to the image-based SVD [28] algorithm. From the computational time perspective, the benefit of the block-based approach to noise level estimation in the SVD domain is significant. For example, compared to the SVD [28] algorithm, the proposed ASVD 64 noise level estimation algorithm reduces the computational time by 50.6%, from 0.0811s to 0.0401s.

The results in Table 5 show that the computational time of a block-based SVD noise level estimation algorithm depends on the size of the image blocks. The highest computational time of 59 ms is attained for SVD 32. For image blocks 64 × 64 and 128 × 128, the computational times of the SVD-B algorithm [29] are 39.6 ms and 40.3 ms, respectively. In consideration of a single image block, the computational time that is required to produce a noise level estimate decreases with the decreasing image block size. However, the computational time is also a function of the number of considered image blocks. The SVD 32 noise level estimation algorithm uses more blocks than SVD 128 algorithm. For a given block size, the computational time can be slightly reduced at the expense of reduced estimation accuracy by decreasing the number of considered blocks. However, this problem is not considered in this paper, but it will be a subject of future research.

The principal objective of this experiment is to determine the computational time cost of adopting the proposed two-stage adaptive noise level estimation compared to a single-stage block-based SVD noise level estimation algorithm in [29]. In another words, is the added computational time cost of the proposed algorithm justified by the improvements in the noise level estimation accuracy levels?

The results in Table 5 show that the proposed ASVD algorithm increases the computational time by less than 1.3% compared to SVD-B [29]. The added computational time does not significantly vary with block size. The reason for only a negligible increase in computational time is attributed to the efficiency of the second-stage noise level estimation. The second stage of the algorithm uses the same image patches and the same sequences of singular values that were evaluated in the first stage of the algorithm. The second stage of the algorithm does not require the singular value decomposition to be performed again.

The time complexity of the algorithm SVD [28] is determined by the time complexity of singular value decomposition of some dense $M \times N$ matrix, $O(M \times N \times \min\{M, N\})$. Time complexity of the singular value decomposition is the principal motivation for the block-based approach to noise level estimation. The time complexity of SVD-B [29] algorithm implemented using the blocks of size $n \times n$, is $O(n^3)$. The time complexity of the proposed ASVD algorithm is also $O(n^3)$. In the second stage of the proposed algorithm, the singular value tail length is adjusted, the associated $P^{(A)}(M)$ and $P^{(B)}(M)$ are evaluated and the new noise level estimate is produced. The time complexity of the second stage of the algorithm is negligible compared to the first stage of the algorithm. The effective design of the proposed method ensures that the added computational time cost compared to the algorithm SVD-B [29] is insignificant.

5. Conclusions

In this paper, an adaptive block-based algorithm for AWGN level estimation in the SVD domain is proposed. The proposed noise level estimation algorithm adopts a variable singular value tail length in order to enable accurate noise level estimation performance at low noise levels. The proposed noise level estimation algorithm is designed to have two distinct stages. In the first stage, a coarse noise level estimate is obtained using the block-based image analysis in the SVD domain, as described in [29]. The coarse noise level estimate is used to adjust the parameters of the algorithm, including the length of singular value tail, in order to produce a more accurate noise level estimate in the second stage of the algorithm. As the AWGN level decreases, the contribution of the underlying image signal to the singular value sequence increases and the number of singular values that are mostly dominated by a noise signal decrease. Thus, the effective singular value tail length is reduced. The ability to reduce the singular value tail length enables the proposed method to significantly improve the noise level estimation accuracy at low noise levels. The paper considers a range of different block sizes, and for each block size, a mathematical model is proposed to describe how to adjust the singular value tail length according to the observed noise levels.

A comparative analysis of the proposed algorithm with the block-based SVD algorithm in [29] and image-based SVD algorithm in [28] is presented. The result show that the proposed noise level estimation algorithm significantly improves the MSE and average error levels across a range of noise levels and for all considered choices of block size. Nevertheless, the proposed noise level estimation algorithm attains the best MSE an average error results for the choice of block size 64 × 64. The only notable drawback of the proposed method is a negligible increase in computational time compared to the algorithm in [29]. This increase in computational time is justified by the significant improvements in MSE and average error values at low noise levels; for example for block size 64 × 64 and AWGN level $\sigma = 1$, the estimation MSE is reduced by more than 65%, whilst the computational time is increased by less than 1.3% The reason why the added computational time is negligible despite requiring two noise level estimates to be produced for each image is attributed to fact that second stage of noise level estimation algorithm uses the same image patches and singular value sequences obtained in the first stage of the algorithm. Thus, singular value decomposition is not performed in the second stage of the algorithm.

The application of the proposed algorithm to the problem of image denoising will probably be the subject of future research work. In addition, improving the computational efficiency of the proposed algorithm based on identification of homogenous image areas and optimizing the number of blocks used in noise level estimation are also possible directions of future research work. The proposed block-based noise level estimation algorithm in the SVD domain can be applied to a variety of digital image processing fields, including image denoising, image segmentation and other problems where a prior knowledge of noise level is required.

Supplementary Materials: The following are available online at http://www.mdpi.com/2079-9292/7/12/397/s1, Figure S1. Standard test images used in the experiments: (**a**) Barbara; (**b**) Boat; (**c**) Cameraman; (**d**) Car; (**e**) Clown; (**f**) Couple; (**g**) Crowd; (**h**) Einstein; (**i**) Girlface; (**j**) Goldhill; (**k**) Lena; (**l**) Baboon; (**m**) Office; (**n**) Peppers; (**o**) Pirate; (**p**) Street; (**q**) Town; (**r**) Woman blonde; (**s**) Woman darkhair; (**t**) Zelda.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A

Table A1. For the block size of 128 × 128, MSE levels across a range of noise levels and normalized singular value tail length k values.

k	**Noise Level, σ**																			
	0.5	**1.0**	**1.5**	**2.0**	**2.5**	**3.0**	**3.5**	**4.0**	**4.5**	**5.0**	**5.5**	**6.0**	**6.5**	**7.0**	**7.5**	**8.0**	**8.5**	**9.0**	**9.5**	**10.0**
0.250	1.050	1.155	0.875	0.932	1.002	1.101	0.675	0.653	0.748	0.677	0.606	0.569	0.657	1.024	0.769	0.589	0.707	0.756	0.470	0.513
0.266	1.291	0.985	0.865	0.862	0.715	0.719	0.849	0.656	0.660	0.579	0.832	0.837	0.533	0.592	0.794	0.614	0.501	0.425	0.552	0.492
0.281	1.177	1.201	0.895	0.868	0.804	0.737	0.679	0.442	0.759	0.795	0.520	0.641	0.587	0.512	0.728	0.508	0.617	0.528	0.596	0.481
0.297	1.200	1.037	0.864	0.672	0.678	0.798	0.653	0.584	0.762	0.556	0.680	0.464	0.662	0.437	0.462	0.516	0.532	0.593	0.460	0.591
0.313	1.271	1.000	0.662	0.797	0.703	0.614	0.841	0.457	0.625	0.613	0.682	0.644	0.498	0.457	0.539	0.518	0.549	0.456	0.557	0.548
0.328	1.093	0.900	0.799	0.854	0.912	0.672	0.756	0.656	0.640	0.567	0.536	0.759	0.549	0.523	0.722	0.668	0.729	0.642	0.505	0.400
0.344	0.963	1.162	0.743	0.789	0.565	0.616	0.561	0.367	0.649	0.474	0.456	0.466	0.396	0.540	0.427	0.381	0.484	0.536	0.487	0.431
0.359	1.151	0.990	0.778	0.934	0.607	0.685	0.643	0.648	0.613	0.767	0.692	0.561	0.604	0.394	0.535	0.563	0.446	0.624	0.517	0.383
0.375	1.173	0.942	0.673	0.720	0.593	0.573	0.718	0.630	0.462	0.525	0.578	0.576	0.525	0.412	0.487	0.458	0.472	0.420	0.327	0.396
0.391	1.110	0.939	0.844	0.741	0.635	0.702	0.540	0.514	0.488	0.569	0.498	0.474	0.426	0.460	0.431	0.413	0.559	0.542	0.439	0.370
0.406	1.140	1.051	0.696	0.784	0.669	0.563	0.575	0.596	0.577	0.445	0.589	0.512	0.553	0.502	0.515	0.401	0.464	0.385	0.447	0.427
0.422	1.198	0.976	0.662	0.658	0.773	0.763	0.533	0.488	0.554	0.540	0.601	0.392	0.501	0.468	0.430	0.530	0.334	0.497	0.373	0.480
0.438	1.108	0.856	0.714	0.822	0.701	0.686	0.623	0.553	0.792	0.484	0.486	0.461	0.373	0.442	0.651	0.438	0.439	0.418	0.428	0.426
0.453	1.080	0.983	0.697	0.750	0.626	0.683	0.527	0.583	0.486	0.566	0.374	0.460	0.458	0.346	0.452	0.329	0.427	0.412	0.354	0.362
0.469	1.121	0.826	0.779	0.652	0.679	0.552	0.632	0.586	0.625	0.470	0.477	0.447	0.476	0.374	0.387	0.397	0.366	0.419	0.406	0.261
0.484	1.390	0.927	0.766	0.610	0.525	0.571	0.361	0.430	0.471	0.486	0.446	0.528	0.357	0.445	0.362	0.333	0.390	0.368	0.234	0.325
0.500	1.198	0.922	0.731	0.637	0.681	0.570	0.608	0.617	0.529	0.458	0.453	0.488	0.454	0.549	0.357	0.429	0.416	0.311	0.463	0.329
0.516	1.179	0.903	0.732	0.655	0.597	0.585	0.578	0.391	0.496	0.583	0.458	0.381	0.456	0.486	0.402	0.357	0.404	0.379	0.385	0.342
0.531	1.306	0.818	0.753	0.553	0.618	0.608	0.486	0.468	0.475	0.599	0.426	0.463	0.475	0.482	0.463	0.453	0.420	0.464	0.337	0.384
0.547	1.254	0.849	0.728	0.596	0.619	0.606	0.581	0.595	0.597	0.505	0.452	0.467	0.456	0.437	0.379	0.402	0.441	0.348	0.349	0.342
0.563	1.364	0.958	0.725	0.713	0.524	0.524	0.495	0.390	0.466	0.422	0.481	0.442	0.327	0.297	0.468	0.299	0.335	0.318	0.235	0.305
0.578	1.423	1.047	0.769	0.675	0.535	0.591	0.481	0.433	0.495	0.455	0.488	0.422	0.322	0.420	0.442	0.448	0.321	0.422	0.340	0.347
0.594	1.460	1.075	0.825	0.613	0.490	0.501	0.475	0.542	0.505	0.395	0.497	0.362	0.367	0.368	0.350	0.326	0.377	0.310	0.210	0.319
0.609	1.558	1.021	0.872	0.572	0.587	0.429	0.481	0.430	0.441	0.505	0.472	0.355	0.469	0.349	0.404	0.305	0.330	0.324	0.279	0.310
0.625	1.680	1.235	0.904	0.688	0.622	0.457	0.448	0.475	0.501	0.409	0.460	0.352	0.358	0.368	0.312	0.296	0.304	0.305	0.284	0.261
0.641	1.725	1.202	0.875	0.754	0.584	0.513	0.481	0.440	0.460	0.414	0.370	0.418	0.369	0.349	0.308	0.316	0.300	0.269	0.335	0.264
0.656	2.288	1.360	1.073	0.881	0.675	0.542	0.518	0.496	0.407	0.421	0.408	0.459	0.313	0.284	0.260	0.303	0.283	0.265	0.209	0.226
0.672	1.881	1.552	1.025	0.743	0.584	0.572	0.417	0.471	0.366	0.390	0.326	0.390	0.294	0.358	0.312	0.324	0.301	0.244	0.243	0.273
0.688	2.345	1.645	1.212	0.843	0.735	0.572	0.498	0.416	0.432	0.425	0.383	0.336	0.327	0.316	0.304	0.272	0.273	0.289	0.249	0.270
0.703	2.855	1.853	1.505	1.094	0.871	0.627	0.537	0.528	0.432	0.387	0.451	0.327	0.385	0.328	0.262	0.280	0.253	0.256	0.289	0.221
0.719	3.111	2.289	1.603	1.135	0.874	0.747	0.556	0.584	0.455	0.464	0.398	0.292	0.362	0.320	0.281	0.270	0.337	0.236	0.250	0.286
0.734	3.256	2.427	1.723	1.406	1.034	0.835	0.737	0.604	0.497	0.463	0.384	0.314	0.356	0.304	0.262	0.328	0.248	0.316	0.235	0.243
0.750	3.769	2.573	1.757	1.389	1.156	0.851	0.748	0.666	0.530	0.469	0.481	0.381	0.394	0.310	0.383	0.292	0.252	0.266	0.253	0.259
0.766	3.822	3.091	2.190	1.892	1.409	1.140	0.831	0.613	0.633	0.557	0.503	0.485	0.391	0.396	0.359	0.366	0.341	0.339	0.320	0.266
0.781	5.042	3.726	3.223	2.417	1.604	1.348	1.207	0.892	0.937	0.673	0.610	0.536	0.522	0.456	0.438	0.472	0.396	0.357	0.354	0.411
0.797	5.725	4.606	3.424	2.460	2.190	1.863	1.382	1.224	1.050	0.920	0.755	0.691	0.511	0.508	0.410	0.507	0.421	0.363	0.386	0.287

Table A2. For the block size of 64 × 64, MSE levels across a range of noise levels and normalized singular value tail length k values.

k	**Noise Level, σ**																			
	0.5	**1.0**	**1.5**	**2.0**	**2.5**	**3.0**	**3.5**	**4.0**	**4.5**	**5.0**	**5.5**	**6.0**	**6.5**	**7.0**	**7.5**	**8.0**	**8.5**	**9.0**	**9.5**	**10.0**
0.250	1.110	1.208	1.045	0.901	0.899	0.912	1.177	0.968	0.777	0.971	0.877	0.979	0.685	0.681	0.892	1.019	0.958	0.963	1.047	1.118
0.266	1.062	0.977	0.765	0.813	0.712	0.622	0.831	0.525	0.586	0.660	0.568	0.704	0.502	0.601	0.784	0.524	0.440	0.590	0.485	0.493
0.281	1.326	1.211	0.877	0.782	0.560	0.599	0.677	0.647	0.418	0.502	0.460	0.504	0.525	0.532	0.364	0.371	0.422	0.362	0.425	0.429
0.297	1.010	0.935	0.564	0.551	0.678	0.645	0.693	0.621	0.479	0.544	0.513	0.536	0.561	0.465	0.660	0.490	0.551	0.612	0.470	0.482
0.313	1.034	0.847	0.738	0.758	0.589	0.661	0.590	0.590	0.710	0.500	0.588	0.437	0.446	0.636	0.413	0.548	0.653	0.435	0.435	0.368
0.328	1.018	0.900	0.795	0.620	0.880	0.920	0.750	0.646	0.768	0.922	0.759	0.680	0.598	0.816	0.628	0.703	0.740	0.639	0.636	0.736
0.344	1.014	0.889	0.748	0.762	0.622	0.522	0.623	0.531	0.503	0.485	0.399	0.418	0.481	0.500	0.392	0.547	0.376	0.467	0.424	0.410
0.359	1.089	0.842	0.655	0.597	0.582	0.531	0.452	0.468	0.662	0.464	0.470	0.523	0.397	0.586	0.402	0.362	0.591	0.422	0.346	0.328
0.375	1.087	0.875	0.585	0.655	0.601	0.735	0.647	0.582	0.518	0.601	0.607	0.558	0.565	0.491	0.555	0.532	0.566	0.418	0.592	0.448
0.391	1.235	0.882	0.762	0.635	0.615	0.614	0.502	0.399	0.339	0.450	0.348	0.452	0.318	0.355	0.345	0.323	0.330	0.364	0.303	0.385

Table A2. *Cont.*

k	Noise Level, σ																			
	0.5	**1.0**	**1.5**	**2.0**	**2.5**	**3.0**	**3.5**	**4.0**	**4.5**	**5.0**	**5.5**	**6.0**	**6.5**	**7.0**	**7.5**	**8.0**	**8.5**	**9.0**	**9.5**	**10.0**
0.406	0.998	0.606	0.729	0.682	0.514	0.535	0.615	0.679	0.502	0.597	0.510	0.348	0.600	0.462	0.495	0.436	0.448	0.363	0.500	0.420
0.422	0.978	0.690	0.607	0.566	0.615	0.514	0.530	0.644	0.436	0.513	0.530	0.565	0.458	0.466	0.384	0.386	0.320	0.506	0.407	0.405
0.438	0.981	0.701	0.691	0.676	0.645	0.623	0.677	0.662	0.615	0.641	0.683	0.651	0.587	0.478	0.538	0.512	0.532	0.598	0.438	0.401
0.453	0.913	0.734	0.804	0.832	0.695	0.870	0.817	0.755	0.648	0.636	0.660	0.842	0.754	0.653	0.626	0.781	0.684	0.666	0.664	0.453
0.469	1.122	0.775	0.671	0.542	0.564	0.557	0.503	0.421	0.563	0.425	0.344	0.388	0.467	0.354	0.304	0.294	0.362	0.339	0.363	0.395
0.484	1.022	0.764	0.678	0.617	0.615	0.714	0.641	0.655	0.613	0.559	0.611	0.634	0.553	0.497	0.642	0.599	0.561	0.524	0.452	0.462
0.500	1.082	0.857	0.641	0.594	0.522	0.445	0.519	0.472	0.442	0.405	0.397	0.386	0.373	0.418	0.340	0.413	0.304	0.359	0.412	0.344
0.516	0.993	0.764	0.718	0.577	0.591	0.520	0.586	0.525	0.474	0.472	0.487	0.596	0.567	0.446	0.469	0.386	0.462	0.479	0.403	0.455
0.531	1.385	0.973	0.591	0.611	0.523	0.579	0.572	0.461	0.468	0.467	0.369	0.418	0.350	0.284	0.320	0.329	0.338	0.327	0.387	0.249
0.547	1.199	0.881	0.644	0.575	0.545	0.428	0.469	0.359	0.331	0.376	0.375	0.358	0.350	0.352	0.373	0.405	0.430	0.341	0.348	0.289
0.563	1.242	0.981	0.691	0.551	0.471	0.472	0.471	0.434	0.471	0.437	0.442	0.382	0.355	0.345	0.370	0.358	0.381	0.330	0.337	0.286
0.578	1.310	0.980	0.716	0.552	0.500	0.551	0.403	0.435	0.347	0.421	0.353	0.366	0.403	0.295	0.339	0.323	0.374	0.342	0.261	0.242
0.594	1.407	0.940	0.674	0.556	0.617	0.501	0.452	0.424	0.506	0.413	0.467	0.399	0.401	0.381	0.439	0.296	0.343	0.294	0.314	0.355
0.609	1.448	1.009	0.732	0.551	0.477	0.438	0.404	0.406	0.410	0.380	0.380	0.399	0.372	0.410	0.356	0.341	0.324	0.342	0.351	0.368
0.625	1.641	1.100	0.774	0.627	0.556	0.469	0.417	0.399	0.328	0.401	0.350	0.352	0.380	0.350	0.317	0.301	0.320	0.260	0.313	0.296
0.641	1.690	1.142	0.754	0.568	0.517	0.459	0.422	0.430	0.369	0.347	0.324	0.377	0.380	0.322	0.416	0.310	0.266	0.274	0.333	0.251
0.656	1.848	1.290	0.953	0.674	0.559	0.440	0.489	0.405	0.364	0.303	0.298	0.330	0.281	0.299	0.292	0.293	0.271	0.222	0.254	0.263
0.672	1.865	1.162	0.847	0.615	0.568	0.456	0.425	0.421	0.351	0.351	0.364	0.304	0.313	0.313	0.334	0.269	0.320	0.248	0.320	0.274
0.688	1.932	1.285	0.961	0.727	0.556	0.459	0.432	0.379	0.330	0.327	0.295	0.329	0.296	0.308	0.232	0.273	0.244	0.272	0.243	0.240
0.703	2.211	1.462	0.983	0.797	0.634	0.471	0.421	0.400	0.401	0.338	0.316	0.367	0.279	0.288	0.264	0.248	0.358	0.274	0.291	0.232
0.719	2.603	1.725	1.193	0.954	0.725	0.515	0.441	0.388	0.314	0.304	0.297	0.257	0.299	0.262	0.253	0.250	0.217	0.240	0.272	0.312
0.734	2.821	2.059	1.589	1.180	0.859	0.647	0.490	0.507	0.330	0.367	0.330	0.286	0.312	0.295	0.229	0.226	0.216	0.231	0.198	0.233
0.750	3.096	1.935	1.379	0.961	0.804	0.676	0.565	0.412	0.380	0.335	0.318	0.292	0.281	0.272	0.257	0.240	0.283	0.238	0.207	0.211
0.766	3.337	2.191	1.639	1.193	0.999	0.715	0.581	0.477	0.348	0.360	0.320	0.278	0.298	0.280	0.237	0.231	0.250	0.226	0.203	0.252
0.781	3.842	2.709	2.048	1.414	1.256	0.897	0.724	0.533	0.544	0.442	0.385	0.291	0.304	0.270	0.229	0.210	0.261	0.191	0.220	0.202
0.797	4.696	3.692	2.540	1.904	1.434	1.203	1.003	0.859	0.639	0.578	0.491	0.439	0.353	0.336	0.276	0.259	0.268	0.264	0.248	0.227

Table A3. For the block size of 32 × 32, MSE levels across a range of noise levels and normalized singular value tail length *k* values.

k	Noise Level, σ																			
	0.5	**1.0**	**1.5**	**2.0**	**2.5**	**3.0**	**3.5**	**4.0**	**4.5**	**5.0**	**5.5**	**6.0**	**6.5**	**7.0**	**7.5**	**8.0**	**8.5**	**9.0**	**9.5**	**10.0**
0.250	1.417	1.072	1.032	0.882	0.768	0.835	0.726	0.770	0.645	0.540	0.616	0.655	0.544	0.591	0.811	0.690	0.648	0.635	0.724	0.755
0.281	1.175	1.072	0.989	0.826	0.944	0.949	0.901	0.917	0.894	0.728	0.739	0.704	0.753	0.714	0.673	0.678	0.724	0.721	0.775	0.667
0.313	1.215	1.165	1.244	1.041	1.193	1.136	1.056	1.051	1.018	1.065	0.880	1.038	0.965	0.841	0.963	0.887	0.856	0.988	0.891	0.772
0.344	1.063	0.946	0.809	0.744	0.804	0.874	0.766	0.699	0.607	0.824	0.807	0.700	0.713	0.748	0.646	0.619	0.617	0.602	0.584	0.686
0.375	1.234	0.867	0.784	0.679	0.678	0.604	0.640	0.523	0.534	0.506	0.423	0.442	0.442	0.401	0.421	0.457	0.421	0.429	0.392	0.332
0.406	1.046	0.989	0.947	0.814	0.846	0.878	0.809	0.772	0.884	0.779	0.799	0.822	0.786	0.772	0.726	0.696	0.767	0.701	0.658	0.610
0.438	1.137	0.863	0.772	0.733	0.612	0.586	0.555	0.618	0.542	0.597	0.517	0.543	0.508	0.514	0.503	0.434	0.484	0.508	0.510	0.477
0.469	1.255	0.901	0.722	0.696	0.598	0.602	0.490	0.547	0.479	0.514	0.463	0.555	0.492	0.479	0.478	0.432	0.343	0.371	0.364	0.378
0.500	1.191	0.882	0.739	0.676	0.592	0.581	0.542	0.522	0.502	0.499	0.480	0.470	0.487	0.507	0.416	0.393	0.423	0.399	0.404	0.375
0.531	1.300	0.926	0.740	0.664	0.648	0.640	0.527	0.510	0.624	0.524	0.533	0.529	0.475	0.483	0.454	0.456	0.431	0.400	0.466	0.416
0.563	1.254	0.932	0.780	0.661	0.593	0.575	0.564	0.550	0.523	0.540	0.521	0.525	0.495	0.523	0.442	0.473	0.433	0.467	0.452	0.424
0.594	1.194	0.894	0.733	0.664	0.633	0.610	0.622	0.653	0.585	0.631	0.613	0.621	0.569	0.557	0.604	0.614	0.609	0.541	0.563	0.544
0.625	1.593	1.111	0.787	0.634	0.596	0.508	0.478	0.458	0.462	0.425	0.411	0.419	0.478	0.390	0.370	0.365	0.383	0.384	0.388	0.355
0.656	1.489	0.993	0.744	0.626	0.578	0.540	0.540	0.515	0.534	0.555	0.532	0.541	0.557	0.543	0.549	0.543	0.529	0.550	0.564	0.487
0.688	1.746	1.125	0.810	0.659	0.574	0.520	0.448	0.461	0.425	0.458	0.485	0.424	0.471	0.498	0.445	0.465	0.439	0.479	0.434	0.460
0.719	2.308	1.622	1.117	0.856	0.647	0.530	0.476	0.412	0.404	0.373	0.359	0.333	0.332	0.344	0.326	0.295	0.293	0.301	0.311	0.333
0.750	2.731	1.943	1.387	0.990	0.728	0.591	0.498	0.396	0.373	0.339	0.312	0.299	0.282	0.277	0.299	0.287	0.309	0.268	0.289	0.335
0.781	2.846	1.988	1.476	1.028	0.780	0.609	0.482	0.405	0.347	0.301	0.289	0.298	0.283	0.280	0.297	0.281	0.302	0.320	0.307	0.325

References

1. Chervyakov, N.; Lyakhov, P.; Kaplun, D.; Butusov, D.; Nagornov, N. Analysis of the quantization noise in discrete wavelet transform filters for image processing. *Electronics* **2018**, *7*, 135. [CrossRef]
2. Turajlic, E.; Karahodzic, V. An Adaptive Scheme for X-ray Medical Image Denoising using Artificial Neural Networks and Additive White Gaussian Noise Level Estimation in SVD Domain. In Proceedings of the International Conference on Medical and Biological Engineering (CMBEBIH 2017), Sarajevo, Bosnia and Herzegovina, 16–18 March 2017; pp. 36–40.
3. Uwe, S.; Kevin, S. Bayesian deblurring with integrated noise estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), Colorado Springs, CO, USA, 20–25 June 2011; pp. 2625–2632.
4. Dabov, K.; Foi, A.; Katkovnik, V.; Egiazarian, K. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.* **2007**, *16*, 2080–2095. [CrossRef] [PubMed]
5. Liu, X.; Tanaka, M.; Okutomi, M. Single-image noise level Estimation for blind denoising. *IEEE Trans. Image Process.* **2013**, *22*, 5226–5237. [CrossRef] [PubMed]
6. Elder, J.H.; Zucker, S.W. Local scale control for edge detection and blur estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 699–716. [CrossRef]

7. Arbelaez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 898–916. [CrossRef] [PubMed]
8. Fu, P.; Sun, X.; Sun, Q. Hyperspectral image segmentation via frequency-based similarity for mixed noise estimation. *Remote Sens.* **2017**, *9*, 1237. [CrossRef]
9. Jan, L.; Fridrich, J.; Goljan, M. Digital camera identification from sensor pattern noise. *IEEE Trans. Inf. Forens. Sec.* **2006**, *1*, 205–214.
10. Greenberg, S.; Aladjem, M.; Kogan, D. Fingerprint image enhancement using filtering techniques. *Real-Time Imaging* **2002**, *8*, 227–236. [CrossRef]
11. Lee, J.S.; Hoppel, K. Noise modeling and estimation of remotely sensed images. In Proceedings of the 12th Canadian Symposium on Remote Sensing Geoscience and Remote Sensing Symposium, Vancouver, BC, Canada, 10–14 July 1989; pp. 1005–1008.
12. Wang, S.; Liu, H.; Xie, K.; Chen, Z.; Zhang, J. Noise level estimation using gradients of image blocks. In Proceedings of the 5th International Conference on Computer Science and Network Technology (ICCSNT 2016), Changchun, China, 10–11 December 2016; pp. 758–761.
13. Huang, X.; Chen, L.; Tian, J.; Zhang, X.; Fu, X. Blind noisy image quality assessment using block homogeneity. *Comput. Electr. Eng.* **2014**, *40*, 796–807. [CrossRef]
14. Huang, X.; Chen, L.; Tian, J.; Zhang, X. Blind image noise level estimation using texture-based eigenvalue analysis. *Multimed. Tools Appl.* **2016**, *75*, 2713–2724. [CrossRef]
15. Abramova, V. A Blind Method for Additive Noise Variance Evaluation Based on Homogeneous Region Detection Using the Fourth Central Moment Analysis. *Telecommun. Radio Eng.* **2015**, *74*, 1651–1669. [CrossRef]
16. Olsen, S.I. Estimation of noise in images: An evaluation. *Graph. Models Image Proc.* **1993**, *55*, 319–323. [CrossRef]
17. Bilcu, R.C.; Vehvilainen, M.A. New Method for Noise Estimation in Images. In Proceedings of the IEEE-EURASIP International Workshop on Nonlinear Signal and Image Processing, Sapporo, Japan, 18–20 May 2005.
18. Tai, S.C.; Yang, S.M. A fast method for image noise estimation using Laplacian operator and adaptive edge detection. In Proceedings of the 3rd International Symposium on Communications, Control, and Signal Processing, St. Julians, Malta, 12–14 March 2008; pp. 1077–1081.
19. Shin, D.; Park, R.; Yang, S.; Jung, J. Block-based noise estimation using adaptive Gaussian filtering. *IEEE Trans. Consum. Electron.* **2005**, *51*, 218–226. [CrossRef]
20. Yang, S.M.; Tai, S.C. Fast and reliable image-noise estimation using a hybrid approach. *J. Electron. Imaging* **2010**, *19*, 033007:1–033007:15. [CrossRef]
21. Turajlic, E. A fast noise level estimation algorithm based on adaptive image segmentation and Laplacian convolution. In Proceedings of the 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2017), Opatija, Croatia, 22–26 May 2017; pp. 486–491.
22. Tang, C.; Yang, X.; Zhai, G. Dual-transform based noise estimation. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2012), Melbourne, Australia, 9–13 July 2012; pp. 991–996.
23. Manjón, J.V.; Coupé, P.; Buades, A. MRI noise estimation and denoising using non-local PCA. *Med. Image Anal.* **2015**, *22*, 35–47. [CrossRef] [PubMed]
24. Ponomarenko, M.; Gapon, N.; Voronin, V.; Egiazarian, K. Blind estimation of white Gaussian noise variance in highly textured images. *J. Electron. Imaging* **2018**, *13*, 382:1–382:5. [CrossRef]
25. Ghazi, M.M.; Erdogan, H. Image noise level estimation based on higher-order statistics. *Multimed. Tools Appl.* **2017**, *76*, 2379–2397. [CrossRef]
26. Chen, G.; Zhu, F.; Heng, A.P. An efficient statistical method for image noise level estimation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV 2015), Santiago, Chile, 7–13 December 2015; pp. 477–485.
27. Khmag, A.; Ramli, A.R.; Al-Haddad, S.A.R.; Kamarudin, N. Natural image noise level estimation based on local Blind image noise level estimation ics for blind noise reduction. *Vis. Comput.* **2018**, *34*, 575–587. [CrossRef]
28. Liu, W.; Lin, W. Additive white Gaussian noise level estimation in SVD domain for images. *IEEE Trans. Image Process.* **2013**, *22*, 872–883. [CrossRef] [PubMed]

29. Liu, W. Additive white Gaussian noise level estimation based on block SVD. In Proceedings of the IEEE Workshop on Electronics, Computer and Applications, Ottawa, ON, Canada, 8–9 May 2014; pp. 960–963.

30. Du, K.L.; Swamy, M.N.S. *Search and Optimization by Metaheuristics*; Birkhäuser: Basel, Switzerland, 2016; pp. 153–173. ISBN 978-33-1941-191-0.