Intelligent SVD-Based Noise Level Estimation Incorporating Symbiotic Organisms Search

Heri Prasetyo[1], Winarno[1], Chih-Hsien Hsia[2]

[1] Department of Informatics, Universitas Sebelas Maret (UNS), Indonesia
[2] Department of Computer Science and Information Engineering, National Ilan University, Taiwan
heri.prasetyo@staff.uns.ac.id; win@staff.uns.ac.id; chhsia625@gmail.com

Abstract

A simple technique for inferring the level of Additive White Gaussian Noise (AWGN) from a still image is presented in this paper. This technique exploits the effectiveness of Singular Value Decomposition (SVD) to estimate the noise level of a noisy image. It investigates the trailing sum of its singular values which contain the noise information of an image. The noise level and two additional parameters own linear dependency with the trailing sum of singular values. The two additional parameters can be experimentally obtained from a given set of noisy images. However, it becomes less satisfied in practical noise level estiation which requires a fast response. Thus, the proposed method utilizes the Symbiotic Organisms Search (SOS) to further optimize the scaling factor, regarded as additional parameter. The extensive experiments show that the proposed method offers a promising result on estimating the noise level. In addition, the estimated noise level can be further employed for the blind image denoising task.

Keywords: Gaussian, Noise estimation, Scalar constant, Symbiotic Organisms Search, Singular value

1 Introduction

The noise level estimation and image denoising algorithms has been attracted so many attentions in recent years [1-7]. The noise level estimation in [1] aims to infer the level of noise from image corrupted with AWGN. It investigates the trailing sum of singular values from noisy image to infer the noise level. While, the image denoising [7] removes the occurred AWGN from noisy image under various strategies. Commonly, the image denoising methods require some prior knowledges about the typical noise information, noise model, the noise probability or level, etc., in order to reduce or remove the occurred noise in a given image. In special task, one only needs to know the noise level in advance for performing the image denoising.

The image denoising with prior noise model and level is less satisfied over several computer vision tasks and image processing applications. In ideal situation, the noise level should be directly estimated from a noisy image. The information about the estimated noise level is very useful for performing the image denoising. Herein, we reach the image denoising process with blind scenario. In this work, we modify the SVD-based noise level estimation of [1]. The proposed method optimizes the parameters of SVD-based estimation [1] with the help of SOS algorithms [8-10]. The proposed method overcomes the difficulty of image denoising module while the information about noise level is unavailable. It can be further extended for multiple secret sharing [11-13], progressive secret sharing [14], vehicle verification [16], and the other image processing and computer vision tasks. The proposed noise level estimation can be applied into some applications under the cloud computing environments such as in [17-19].

2 SVD-Based Noise Level Estimation

This section briefly reviews the former SVD-based noise level estimation [1]. The SVD is very effective tools to decompose an arbitrary matrix into three different matrices. The noise level of a noisy image can be easily observed and inferred by means of SVD operation. Let I be a noise-free (or clean) grayscale image, and A be a noisy version of image I. In this paper, this noisy image is simply modelled as follow:

$$A = I + \mathcal{N}(0, \sigma^2), \quad \textbf{(1)}$$

where A denotes the noisy image, and $\mathcal{N}(0, \sigma^2)$ is AWGN with specific noise level σ. Our goal is to estimate the noise level $\hat{\sigma}$, from a noisy image A. The estimated noise level should be as closest as possible to the original level, indicated with $\hat{\sigma} \approx \sigma$.

For estimating $\hat{\sigma}$, a noisy image A is firstly decomposed using SVD operation as follow:

$$A \Rightarrow U_A \Sigma_A V_A^T, \quad \textbf{(2)}$$

*Corresponding Author: Chih-Hsien Hsia; E-mail: chhsia625@gmail.com
DOI: 10.3966/160792642021012201006

where U_A, Σ_A, and V_A are left singular vectors, singular value matrix, and right singular vectors, respectively. These three matrices own interesting properties such as $U_A U_A^T = U_A^T U_A = I$ and $V_A V_A^T = V_A^T V_A = I$, as well as $\Sigma_A = diag\{\lambda_1^A, \lambda_2^A, ..., \lambda_r^A, 0, ..., 0\}$ with condition $\lambda_1^A \geq \lambda_2^A \geq \cdots \geq \lambda_r^A > 0$. The symbol λ_i^A denotes the *i*-th singular value of noisy image A. Herein, the symbol r refers to the rank of matrix A. As proven in [1], the matrix Σ_A contains the information about the noise level. Thus, the noise level can be simply inferred from Σ_A under some extends.

We firstly calculate the sum of trailing singular values from noisy image A as follow:

$$\eta_A(M) = \frac{1}{M} \sum_{i=r-M+1}^{r} \lambda_i^A, \tag{3}$$

where M denotes the number of considered singular values for estimating $\hat{\sigma}$, and $\eta_A(M)$ is the sum of trailing singular values of A. The value of M should be chosen as $1 \leq M \leq r$. W. Liu et.al. [1] suggests a good choice for M as $M = \frac{3}{4}r$. Figure 1 shows the training images used in the experiment, while Figure 2 depicts the relationship between $\eta_A(M)$ and σ. It can be observed from Figure 2, the values of $\eta_A(M)$ and σ have the linear relationship as follow:

$$\eta_A(M) = a\sigma. \tag{4}$$

where a denotes a scalar constant. If one obtains a, the value of $\hat{\sigma}$ can be directly computed using (4). Herein, the constant a can be computed using the Least Squared Fitting (LSF) method by involving a set of training images.

The calculation in (4) gives low accuracy on estimated noise level. To obtain better estimation, the noisy image A should be injected with additional AWGN as previously used in [1]. This process is conducted as follow:

$$B \Leftarrow A + \mathcal{N}(0, \sigma_B^2), \tag{5}$$

where B is the doubled noisy image, $\mathcal{N}(0, \sigma_B^2)$ denotes the AWGN process with known noise level σ_B^2. The matrix B is further decomposed using SVD to yield the following result:

$$B \Rightarrow U_B \Sigma_B V_B^T, \tag{6}$$

where U_B and V_B are two unitary matrices satisfying $U_B U_B^T = U_B^T U_B = I$ and $V_B V_B^T = V_B^T V_B = I$. While $\Sigma_B = diag\{\lambda_1^B, \lambda_2^B, ..., \lambda_r^B, 0, ..., 0\}$ is singular value matrix. The sum of trailing singular values of B, denoted as $\eta_B(M)$, is then computed as follow:

Figure 1. A set of training images used in the experiment

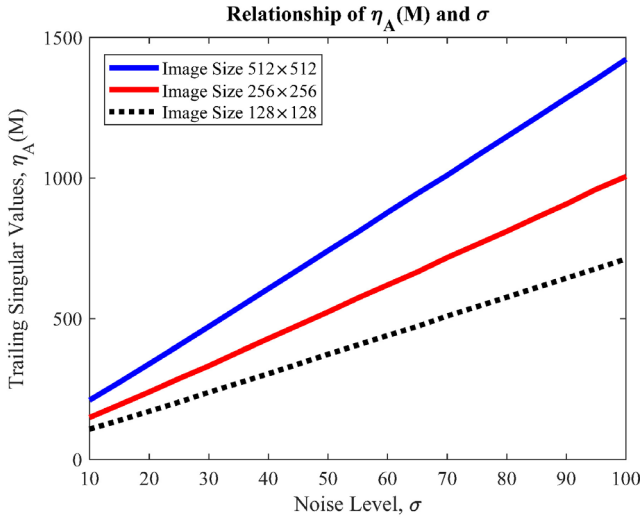


Figure 2. Relationship between the trailing singular values of A, $\eta_A(M)$, and noise level σ

$$\eta_A(M) = \frac{1}{M} \sum_{i=r-M+1}^{r} \lambda_i^B. \tag{7}$$

The value of M is again determined as $M = \frac{3}{4}r$, while r is matrix rank of B. Figure 3 draws the relationship between $\eta_B(M)$ and σ. Now, the relationship between these two values cannot be easily fitted with simple linear function as indicated in (4).

The remedial actions should be carried out to achieve better fitting for $\eta_A(M)$, $\eta_B(M)$, σ, and σ_B. Herein, we incorporate an additional parameter β along with the usage of a. The following relations offer better fitting result for the noisy image:

$$\begin{cases} \eta_B(M) = a\sqrt{\sigma^2 + \sigma_B^2} + \beta \\ \eta_A(M) = a\sigma + \beta \end{cases}. \tag{8}$$

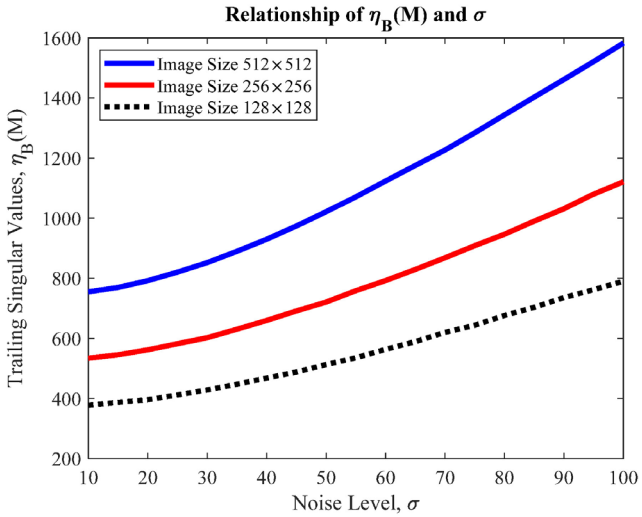


Figure 3. Relationship between the trailing singular values of B, $\eta_B(M)$, and noise level σ

The LSF cannot directly solve (8) in order to obtain the optimum values of a and β. With a simple substitution on β, we obtain the following result:

$$\eta_A(M) - a\sigma = P_B(M) - a\sqrt{\sigma^2 + \sigma_B^2}. \quad \textbf{(9)}$$

Continuing with little algebraic manipulations, the formulation in (9) can be further recomputed as follow:

$$\begin{aligned}
&a\sqrt{\sigma^2+\sigma_B^2} - a\sigma = \eta_B(M) - \eta_A(M),\\
&\{\frac{1}{a}\{\eta_B(M)-\eta_A(M)\}+\sigma\}^2 = \sigma^2+\sigma_B^2,\\
&\frac{2\sigma}{a}\{\eta_B(M)-\eta_A(M)\}+\sigma^2+\frac{1}{a^2}\{\eta_B(M)-\eta_A(M)\}^2\\
&=\sigma^2+\sigma_B^2,\\
&\frac{2\sigma}{a}\{\eta_B(M)-\eta_A(M)\} = \sigma_B^2 - \frac{1}{\sigma^2}\{\eta_B(M)-\eta_A(M)\}^2. \quad \textbf{(10)}
\end{aligned}$$

Dividing the term $\{\eta_B(M)-\eta_A(M)\}$ for both side in (10), the last form in (10) can be further simplified as follow:

$$\frac{2\sigma}{a} = \frac{\sigma_B^2}{\{\eta_B(M)-\eta_A(M)\}} - \frac{1}{a^2}\{\eta_B(M)-\eta_A(M)\}. \quad \textbf{(11)}$$

Finally, the noise level can be estimated using the following formula:

$$\hat{\sigma}_{SVD} = \frac{a\sigma_B^2}{2[\eta_B(M)-\eta_A(M)]} - \frac{\eta_B(M)-\eta_A(M)}{2a}, \quad \textbf{(12)}$$

where $\hat{\sigma}_{SVD}$ is the noise level of A under a predetermined a as used in [1]. As reported in [1], the SVD-based noise level estimation yields a good result by applying an additional AWGN with noise level σ_B^2.

3 SOS-Based Noise Level Estimation

This section presents the proposed noise level estimation. This proposed scheme inherits the effectiveness of SVD-based noise level estimation [1] for inferring noise level. The main difference between our proposed method and former scheme [1] is on determination of scaling factor a. The value of a is from LSF method over several images set. Whereas, the proposed method searches the optimum scaling factor a with the aid of SOS algorithm from a set of training images.

3.1 SOS Algorithm

The SOS mimics the behavior natures, i.e. symbiotic organism activity, for performing the numerical optimization. This algorithm searches an optimum solution over specific search space. In contrast with the other nature-inspired algorithms, the SOS has advantages on free parameters setting and easy to implement. The basic SOS consists of three different phases, i.e. mutualism, commensalism, and parasite phase. Let $X = \{X_1, X_2, \ldots, X_N\}$ be a solution candidate or population, where N denotes the size of ecosystem. The i-th organism is firstly initialized with $X_i = U(a,b)$, where $U(a,b)$ is number generator producing uniformly random number over lower bound a and upper bound b. Suppose that each organism consists of D dimension. At the initial process, we identify the best solution X^{best} from the initial solution. For each organism X_i, we perform the following process for $i = 1, 2, \ldots, N$:

A **mutualism phase** describes the relationship between two organisms while these two organisms share the reciprocally benefit. In the SOS computation, we randomly select index j while $j = \{1, 2, \ldots, N\}$ with constraint $j \neq i$. This condition implies $X_i \neq X_j$. The mutual organism, i.e. X^{mutual}, can be obtained as follow:

$$X^{mutual} = \frac{1}{2}(X_i + X_j). \quad \textbf{(13)}$$

This mutual organism induces a new couple of organisms, referred as X_i^{new} and X_j^{new}. These two new organisms consider the best solution found so far X^{best} and the mutual organism X^{mutual}. The determination of these two new organisms can be formally defined as follow:

$$X_i^{new} = X_i + U(0,1)[X^{best} - \gamma_1 X^{mutual}], \quad \textbf{(14)}$$

$$X_j^{new} = X_j + U(0,1)[X^{best} - \gamma_2 X^{mutual}], \quad \textbf{(15)}$$

where γ_1 and γ_2 are two constants representing the contribution of mutual organism on calculating X_i^{new} and X_j^{new}. These two constants can be simply computed as follow:

$$\gamma_1 = 1 + \text{Round}\{U(0,1)\}, \quad \textbf{(16)}$$

$$\gamma_2 = 1 + \text{Round}\{U(0,1)\}, \quad \textbf{(17)}$$

where $U(0,1)$ is uniform random number in range $[0,1]$. In case of function minimization, the value of X_i is replaced with X_i^{new} if the new organism X_i^{new} offers a better solution. This replacement is also conducted for X_j. The replacement process can be explicitly specified using the following rule based system:

$$\begin{cases} \text{IF } f(X_i^{new}) < f(X_i), \textit{Then } X_i = X_i^{new} \\ \text{IF } f(X_j^{new}) < f(X_j), \textit{Then } X_j = X_j^{new} \end{cases}, \quad \textbf{(18)}$$

where $f(\cdot)$ denotes the fitness function from a given organism. The SOS performs mutualism phase for all organisms.

The **Commensalism phase** resembles the interaction between two organisms. The first organism strives its benefits from the other organism, while the other organism may receive or not this kind of benefit. The process of commensalism phase can be described as follow. Firstly, the SOS randomly selects the organism X_j, where j is randomly chosen from $\{1, 2, ..., N\}$ with constraint $j \neq i$. The new organism from commensalism phase, X_i^{new} is simply determined based on the values of X_i, X_j, and the best solution found so far X^{best}. The following describes this process:

$$X_i^{new} = X_i + U(-1,1)[X^{best} - X_j]. \quad \textbf{(19)}$$

The SOS replaces the current organism X_i with X_i^{new}, if X_i^{new} offers better solution compared to that of X_i. The replacement process for function minimization is formally defined as follow:

$$\text{IF } f(X_i^{new}) < f(X_i), \text{Then } X_i = X_i^{new}. \quad \textbf{(20)}$$

The commensalism phase is repeated over all organisms X_i, for $i = 1, 2, ..., N$.

The **Parasitism phase** is the last stage on the SOS optimization. This phase is almost similar to the nature of symbiotic. One organism may derives many benefits by harming or killing another organism. Herein, an attacked organism is regarded as parasite $X^{parasite}$. The process of parasitism phase can be formulated as follow:

$$X^{parasite} = \begin{cases} U(0,1) \geq 0.5 \\ a + U(0,1)[b-a], else \end{cases}. \quad \textbf{(21)}$$

The next step, we need to determine the organism X_j, where j is randomly chosen from $j = 1, 2, ..., N$ under the constraint $j \neq i$. The organism X_j is simply replaced with $X^{parasite}$ under the following condition:

$$\text{IF } f(X^{parasite}) < f(X_j), \text{Then } X_j = X^{parasite}. \quad \textbf{(22)}$$

The SOS repeats these operations over several iterations or until a specific stopping criteria is met. The implementation of SOS does not require some tuning parameters. Thus, the SOS can be directly implemented without user predetermined parameters.

3.2 Noise Level Estimation Using SOS

This subsection presents our proposed method for SVD-based noise level estimation incorporating SOS. Let $S = \{A_1, A_2, ..., A_T\}$ be a set of training images, while A_i is noisy image, for $i = 1, 2, ..., T$. The noise levels for each noisy image are recorded as a set of noise level $\sigma_{GT} = \{\sigma_1, \sigma_2, ..., \sigma_T\}$, where σ_i denotes the noise level corresponding to noisy image A_i. The noisy image and its noise level can be regarded as training set for computing the optimum a^* via SOS optimizer.

The SOS iteratively optimizes the value of a by encoding its individual organism as a candidate of optimum scaling value a, i.e. $X = a$ for $i = 1, 2, ..., N$. Herein, each organism is firstly initialized as $X_i \sim U(a, b)$, while a and b are the lower and upper bound of search space, respectively. The fitness function of SOS for SVD-based noise level estimation can be computed by measuring the difference between the estimated noise level from SOS and the ground truth noise level. Herein, the noise level is estimated by decoding the value of X_i as a for estimating $\hat{\sigma}$. The difference between the real and estimated noise level can be simply measured under the Manhattan (l_1) or Euclidean (l_2) distance metric. The l_1 distance is formally defined as follow:

$$\mathcal{L}_1(S_A, \sigma_{GT}, X_i) = \frac{1}{T}\sum_{t=1}^{T} |\sigma_t - \sigma_t; X_i|. \quad \textbf{(23)}$$

Whereas, the l_2 distance is given as bellow:

$$\mathcal{L}_2(S_A, \sigma_{GT}, X_i) = \frac{1}{T}\sum_{t=1}^{T} (\sigma_t - \sigma_t; X_i)^2. \quad \textbf{(24)}$$

After performing SOS optimization over several iterations, we may obtain an optimum organism denoted as X^*. We then decode this optimum organism as the optimum scalar value a as $a_{sos}^* = X^*$,

while a^*_{sos} denotes an optimum scalar value for determining the noise level. Finally, the noise level can be simply inferred by performing the following computation:

$$\hat{\sigma}_{SVD} = \frac{a^*_{sos}\sigma_B^2}{2[\eta_B(M) - \eta_A(M)]} - \frac{\eta_B(M) - \eta_A(M)}{2a^*_{sos}}. \quad \textbf{(25)}$$

From this last form, one may simply estimate the noise level of noisy image by firstly computing trailing singular values $\eta_A(M)$ and $\eta_B(M)$ with the help of optimum a^*_{sos}. The noise level estimation in (25) is trivially identical to that of (12) except on the value of a and a^*_{sos}.

4 Experimental Results

This section explicitly reports some extensive experimental results on the SVD-based noise level estimation with SOS optimizer. We consider two sets as a set of training and testing images, as shown in Figure 1 and Figure 4, respectively. We examine the performance of the proposed method in terms of accuracy on the estimated noise level.

Figure 4. A set of testing images used in the experiment

4.1 Performance of SOS

This subsection evaluates the performance of SOS on finding the optimum value of a^*. Each image in training set is scaled to 128×128, 256×256, and 512×512. Since different image size requires different $a*$. For each training image of Figure 1, we give AWGN over various noise levels, i.e. $\sigma_{GT} = \{10, 20, ..., 100\}$. The value of M is simply set as $M = \frac{3}{4}r$. We simply set the size of ecosystem as $N = 50$. The lower and upper bounds for each organism are simply set as $a = -500$ and $b = 500$, respectively. We give large search space for each SOS organism to further investigate the SOS ability on finding the global optimum. Herein, the maximum function evaluation is simply set as 300. The l_1 or l_2 are used to compute the SOS fitness functions. The SOS avoids some complicated procedures on determining its parameters.

In initial experiment, we conduct the SOS training over 30 independent runs. We then simply find the best and worst run of SOS training on these 30 runs. Figure 5 shows the SOS convergence history between the best and worst run over various image sizes. As shown from this figure, the proposed method can able to obtain the optimum scaling factors a with convergence results after several iterations. In addition, Table 1 delivers the minimum, maximum, mean, median, and standard deviation of fitness function over 30 independent runs and different image sizes. This table indicates that the proposed method achieves stable results on determining the AWGN level. While Table 2 shows the optimum a^* obtained from SVD, LFS, and proposed method with l_1 and l_2. Different image size needs different a^*.

4.2 Performance of Noise Level Estimation

In this subsection, we further investigate the performance of proposed method on estimating the noise level. Herein, we use six grayscale images for testing images as shown in Figure 4. We consider the Mean Absolute Error (MAE) to measure the performance of noise level estimation. Tables 3 and 4 tabulate the average and standard deviation of MAE, respectively. As tabulated from these two tables, the proposed method yields better performance compared to the former existing schemes as indicated with the lowest average and lowest standard deviation of MAE. The proposed method is very competitive for the AWGN level estimation task.

4.3 Performance of Image Denoising

This subsection reports the proposed method performance for blind image denoising. We firstly investigate the performance under visual investigation. Herein, we select one image from testing set and inject AWGN with $\sigma = 35$. The image set is 512×512. We perform the image denoising with BM3D algorithm under the estimated noise level $\hat{\sigma}$. Figure 6 depicts the results of blind image denoising with estimated noise levels from various methods. The proposed method offers the best performance compared to the other schemes.

Table 5 and Table 6 summarize the performance between the proposed method and former schemes on blind image denoising with various noise level estimations. The performance is assessed under the Peak-Signal-to-Noise-Ratio (PSNR) and Structural Similarity Index Metric (SSIM). From Tables 5 and 6, it can be concluded that the proposed method outperforms the former existing schemes.

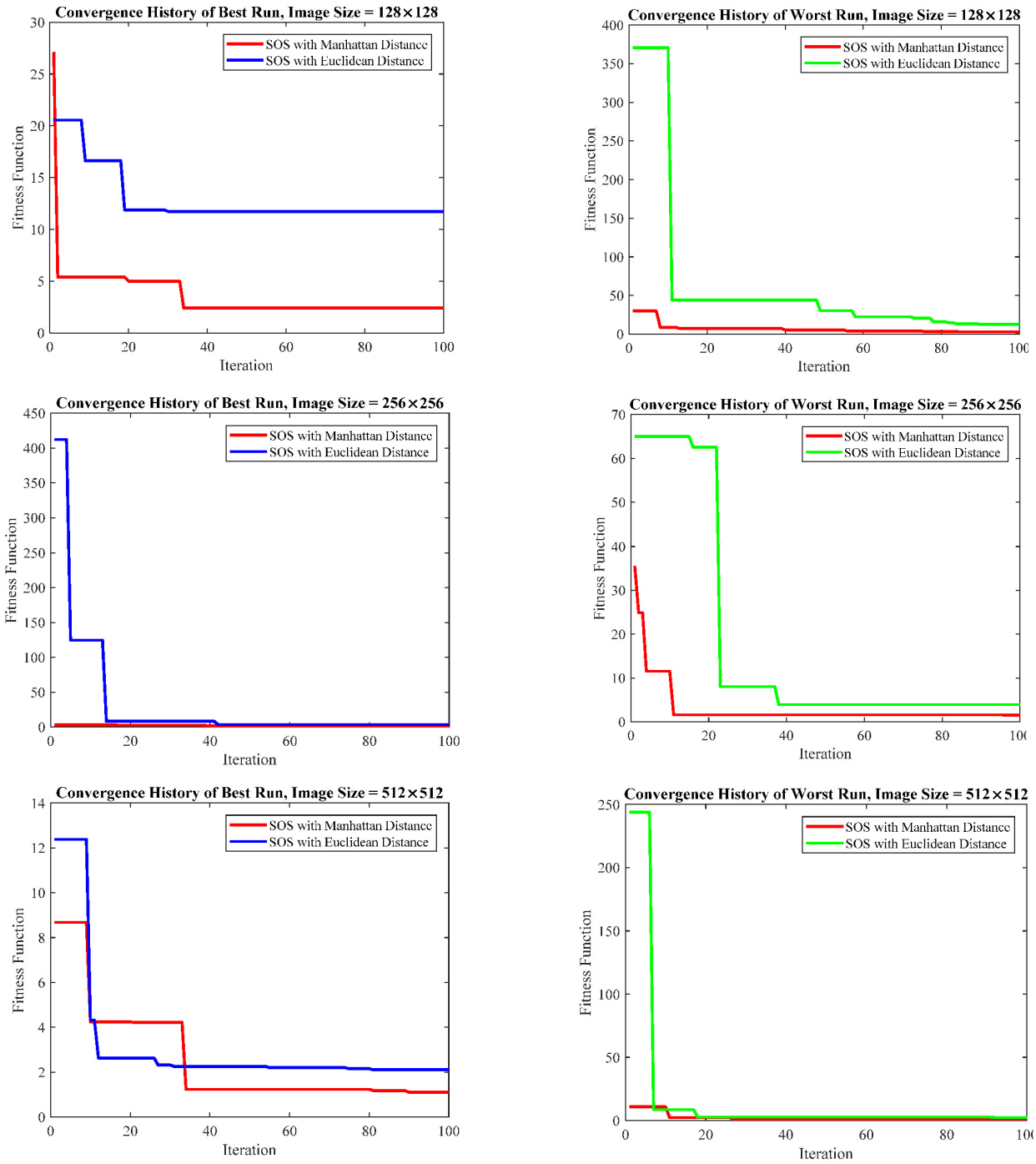


Figure 5. The SOS convergence history of the best run (left column) and worst run (right column) over various image sizes

Table 1. The optimum SOS fitness values over l_1 and l_2 distances

Image Size 128×128					
Method	Minimum	Maximum	Mean	Median	Standard Deviation
SOS l1	2.4079	2.67985	2.44339	2.41811	0.05805
SOS l2	11.7283	12.8042	11.8738	11.7651	0.26837
Image Size 256×256					
Method	Minimum	Maximum	Mean	Median	Standard Deviation
SOS l1	1.39966	1.54601	1.42488	1.40383	0.04424
SOS l2	3.52463	3.94007	3.5958	3.54159	0.11591
Image Size 512×512					
Method	Minimum	Maximum	Mean	Median	Standard Deviation
SOS l1	1.10476	1.16577	1.11862	1.10839	0.01836
SOS l2	2.11215	2.35722	2.13479	2.12043	0.04539

Table 2. The Optimum *a**

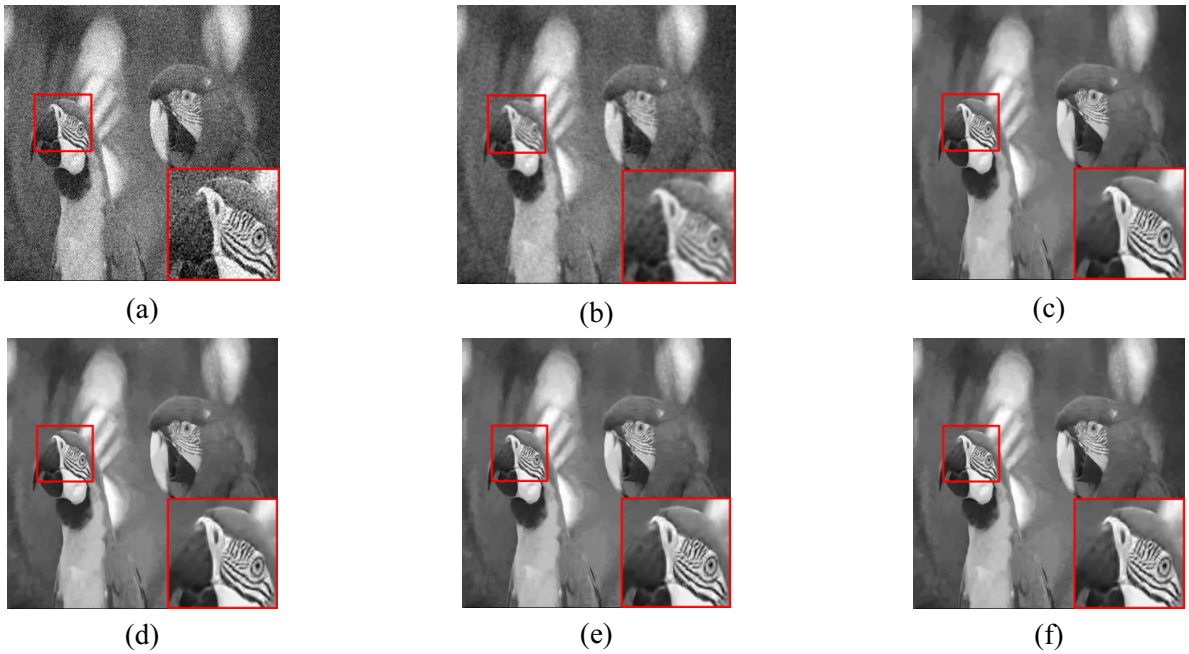
Method	128×128	256×256	512×512
SVD [1]	7.02	9.83	13.87
LFS	7.27113	7.27113	14.4898
SOS l1	6.79818	9.59623	13.6247
SOS l2	6.80497	9.589	13.5716

Table 3. Performance Comparisons in Terms of average MAE

Noise Level	Image Size 128×128				Image Size 256×256				Image Size 512×512			
	LSF	SVD [1]	SOS l_1	SOS l_2	LSF	SVD [1]	SOS l_1	SOS l_2	LSF	SVD [1]	SOS l_1	SOS l_2
10	4.22239	2.27607	1.04607	1.01344	4.07195	1.84341	0.96425	0.95019	4.2059	1.94858	1.2735	1.19178
20	3.88005	1.90479	0.80014	0.7424	3.74998	1.47267	0.50836	0.4996	3.93991	1.53868	0.79519	0.69209
30	3.98011	1.85789	0.97972	0.82578	3.89563	1.34996	0.51362	0.51353	4.046	1.46352	0.63072	0.53864
40	4.54152	2.23603	1.37749	1.17733	4.14368	1.44478	0.71555	0.65728	4.21487	1.42352	0.53569	0.4611
50	4.81934	2.49507	1.7826	1.6327	4.42478	1.48167	0.84518	0.88371	4.54003	1.42786	0.54034	0.54532
60	5.25271	2.6545	2.0836	2.04169	4.88279	1.63281	1.07805	1.16872	4.94867	1.47557	0.64912	0.64503
70	5.64667	3.36364	2.50681	2.77317	5.37472	1.70762	1.46013	1.3403	5.33185	1.50275	0.77664	0.76203
80	7.02301	3.81929	3.36252	3.35913	5.63505	2.17497	1.99977	2.04147	5.70581	1.4716	0.80173	0.97004
90	8.04458	5.6482	4.13268	4.24311	6.01423	2.86148	2.25465	2.18495	6.08622	1.81231	1.00387	1.21584
100	7.81542	5.61692	5.16468	5.2246	6.89329	2.74613	2.53999	2.73969	6.53988	1.85188	1.47754	1.30263
Average	5.52258	3.18724	2.32363	**2.30333**	4.90861	1.87155	**1.28795**	1.29794	4.95591	1.59163	0.84843	**0.83245**

Table 4. Performance Comparisons in Terms of standard deviation MAE

Noise Level	Image Size 128×128				Image Size 256×256				Image Size 512×512			
	LSF	SVD [1]	SOS l_1	SOS l_2	LSF	SVD [1]	SOS l_1	SOS l_2	LSF	SVD [1]	SOS l_1	SOS l_2
10	0.95748	0.95887	0.6778	0.61475	0.90094	0.91839	0.54683	0.54656	1.16636	1.11667	0.82368	0.78532
20	0.94619	0.88203	0.59825	0.55179	0.63234	0.60824	0.37609	0.36192	0.74984	0.733	0.51514	0.48002
30	1.05906	1.06706	0.68725	0.677	0.64175	0.72101	0.42743	0.42009	0.65301	0.64839	0.45197	0.39717
40	1.54772	1.40151	1.0052	0.86214	0.85146	0.72998	0.50157	0.51036	0.59778	0.58366	0.38646	0.32141
50	2.25949	1.78096	1.32582	1.20305	1.02628	0.91684	0.69464	0.63018	0.6789	0.64522	0.41902	0.41268
60	2.83235	1.95938	1.55177	1.45432	1.4833	1.09266	0.72966	0.87313	0.76028	0.74123	0.531	0.48383
70	3.57812	2.19587	2.12955	2.07063	1.73192	1.24284	1.07877	1.05809	0.8885	0.84422	0.57604	0.5993
80	4.30063	2.75177	2.63202	2.58925	2.15303	1.49124	1.33267	1.32256	1.09452	0.93263	0.65282	0.73172
90	4.62976	4.41744	3.01184	2.80959	2.63802	2.26027	1.61456	1.69786	1.46834	1.24199	0.80193	0.86732
100	5.9371	4.5167	3.70366	3.77747	2.97545	2.29202	1.89554	1.93444	1.85668	1.33594	1.13442	0.99156
Average	2.80479	2.19316	1.73232	**1.661**	1.50345	1.22735	**0.91978**	0.93552	0.99142	0.8823	0.62925	**0.60703**



(a) (b) (c) (d) (e) (f)

Figure 6. Examples of blind image denoising on (a) noisy image with $\sigma = 35$. While (b) denoised image with the Gaussian filtering 11×11and 1.5, (c) BM3D with $\hat{\sigma}_{LSF}$, (d) BM3D with $\hat{\sigma}_{SVD}$, (e) BM3D with $\hat{\sigma}_{SOS}$, and (f) ground truth image.

Table 5. Performance Comparisons on Blind Image Denoising in Terms of PSNR

Method	$\sigma = 25$			$\sigma = 55$			$\sigma = 75$		
	128×128	256×256	512×512	128×128	256×256	512×512	128×128	256×256	512×512
Gaussian Filtering	24.5312	25.0893	25.1049	23.0052	23.3889	23.4139	21.7783	22.0756	22.1707
LSF	27.2522	28.1494	28.7061	23.9683	24.8885	25.5118	22.8575	23.7573	24.3823
SVD [1]	27.4391	28.3776	28.9574	24.0950	24.9927	25.5999	22.8558	23.8132	24.4288
ASVD [15]	27.4481	28.3981	28.9912	24.0997	25.0071	25.6001	22.8712	23.8188	24.4331
SOS l_1	27.5675	28.4723	29.0147	24.1258	25.0123	25.6464	22.9126	23.8256	24.4480
SOS l_2	**27.5692**	**28.4881**	**29.0517**	**24.1588**	**25.0216**	**25.6539**	**22.9410**	**23.8361**	**24.4522**

Table 6. Performance Comparisons on Blind Image Denoising in Terms of SSIM

Method	$\sigma = 25$			$\sigma = 55$			$\sigma = 75$		
	128×128	256×256	512×512	128×128	256×256	512×512	128×128	256×256	512×512
Gaussian Filtering	0.67845	0.68608	0.68151	0.57019	0.5594	0.54234	0.50054	0.47801	0.45909
LSF	0.78396	0.7942	0.79693	0.64735	0.67302	0.68899	0.59088	0.61948	0.64260
SVD [1]	0.79412	0.80412	0.80766	0.65752	0.67823	0.69253	0.58961	0.61925	0.64319
ASVD [15]	0.79503	0.80609	0.80888	0.65578	0.67772	0.69354	0.58989	0.61933	**0.64332**
SOS l_1	**0.80032**	0.80739	0.81014	0.65836	**0.67915**	**0.69469**	0.59586	0.61867	0.64243
SOS l_2	0.79988	**0.80843**	**0.81164**	**0.65871**	0.67910	0.69416	**0.59679**	**0.62065**	0.64238

5 Conclusions

A new intelligent SVD-based noise level estimation has been presented in this paper. This new scheme involves the SOS optimizer on finding the optimum scaling factor. In addition, the proposed method inherits the successfully SVD-based noise level estimation. The Experimental Results gives the documentation of the proposed method superiority compared to that of the former scheme in the AWGN level estimation.

References

[1] W. Liu and W. Lin, Additive White Gaussian Noise Level Estimation in SVD Domain for Images, *IEEE Transactions on Image Processing*, Vol. 22, No. 3, pp. 872-883, March, 2013.

[2] Q. Guo, C. Zhang, Y. Zhang, and H. Liu, An Efficient SVD-based Method for Image Denoising, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 26, No. 5, pp. 868-880, May, 2016.

[3] Y.-M. Huang, H.-Y. Yan, Y.-W. Wen, and X. Yang, Rank minimization with Applications to Image Noise Removal, *Information Sciences*, Vol. 429, pp. 147-163, March, 2018.

[4] S. Gu, L. Zhang, W. Zuo, and X. Feng, Weighted Nuclear Norm Minimization with Application to Image Denoising, *Conference on Computer Vision and Pattern Recognition*, Ohio, United States of America, 2014, pp. 2862-2869.

[5] H. Hu, J. Froment, and Q. Liu, A Note on Patch-based Low-Rank Minimization for Fast Image Denoising, *Journal of Visual Communication and Image Representation*, Vol. 50, pp. 100-110, January, 2018.

[6] H. Prasetyo and C.-H. Hsia, Blind Image Denoising Using Low Rank Matrix Minimization, *IEEE International Conference on Consumer Electronics-Taiwan 2018 (IEEE ICCE-TW 2018)*, Taichung, Taiwan, 2018, pp. 1-2.

[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, Image Denoising by Sparse 3D Transform-Domain Collaborative Filtering, *IEEE Transactions on Image Processing*, Vol. 16, No. 8, pp. 2080–2095, August, 2007.

[8] M.-Y. Cheng and D. Prayogo, Symbiotic Organisms Search: a New Metaheuristic Optimization Algorithm, *Computers and Structures*, Vol. 139, pp. 98-112, July, 2014.

[9] A. E. Ezugwu and D. Prayogo, Symbiotic Organisms Search Algorithm: Theory, Recent Advances and Applications, *Expert Systems with Applications*, Vol. 119, pp. 184-209, April, 2019.

[10] S. Al-Sharhan and M. G. Omran, An Enhanced Symbiosis Organisms Search Algorithm: an Empirical Study, *Neural Computing and Applications*, Vol. 29, No. 11, pp. 1025-1043, June, 2018.

[11] H. Prasetyo and J.-M. Guo, A Note on Multiple Secret Sharing Using Chinese Remainder Theorem and Exclusive-OR, *IEEE Access*, Vol. 7, pp. 37473-37497, March, 2019.

[12] H. Prasetyo and C.-H. Hsia, Improved Multiple Secret Sharing Using Generalized Chaotic Image Scrambling, *Multimedia Tools and Applications*, Vol. 78, No. 20, pp. 29089-29120, October, 2019.

[13] J.-M. Guo, D. Riyono, and H. Prasetyo, Improved Beta Chaotic Image Encryption for Multiple Secret Sharing, *IEEE Access*, Vol. 6, pp. 46297-46321, August, 2018.

[14] H. Prasetyo and C.-H. Hsia, Lossless Progressive Secret Sharing for Grayscale and Color Images, *Multimedia Tools and Applications*, Vol. 78, No. 17, pp. 24837-24862, September, 2019.

[15] E. Turajlic, Adaptive SVD Domain-Based White Gaussian Noise Level Estimation in Images, *IEEE Access*, Vol. 6, pp. 72735-72747, November, 2018.

[16] J.-M. Guo, H. Prasetyo, M. E. Farfoura, and H. Lee, Vehicle

Verification using Features from Curvelet Transform and Generalized Gaussian Distribution Modelling, *IEEE Transactions on Intelligent Transportation Systems*, Nol. 16, No. 4, pp. 1989-1998, August, 2015.

[17] B. Kang, J. Lee, O. Bagdasar, and H. Choo, CloudIoT-based Jukebox Platform: A Music Player for Mobile Users in Cafe, *Journal of Internet Technology*, Vol. 21, No. 5, pp. 1363-1374, September, 2020.

[18] J.-S. Li, I.-H. Liu, C.-Y. Lee, C.-F. Li, and C.-G. Liu, A Novel Data Deduplication Scheme for Encrypted Cloud Databases, *Journal of Internet Technology*, Vol. 21, No. 4, pp. 1115-1125, July, 2020.

[19] R. Senthilkumar and B. G. Geetha, Asymmetric Key Blum-Goldwasser Cryptography for Cloud Services Communication Security, *Journal of Internet Technology*, Vol. 21, No. 4, pp. 929-939, July, 2020.

Biographies

Heri Prasetyo received his Bachelor's Degree from the Department of Informatics Engineering, Institut Teknologi Sepuluh Nopember (ITS), Indonesia in 2006. He received his Master's and Doctoral Degrees from the Department of Computer Science and Information Engineering, and Department of Electrical Engineering, at the National Taiwan University of Science and Technology (NTUST), Taiwan, in 2009 and 2015, respectively. He received the Best Dissertation Award from the Taiwan Association for Consumer Electronics in 2015, Best Paper Awards from the International Symposium on Electronics and Smart Devices 2017, ISESD 2019, and International Conference on Science in Information Technology 2019, and the Outstanding Faculty Award 2019 from his current affiliated university. His research interests include multimedia signal processing, computational intelligence, pattern recognition, and machine learning.

Winarno received his Bachelor's Degree from the Department of Mathematics, Universitas Sebelas Maret (UNS), Indonesia, in 2005. He received his Master's Degree from the Department of Electrical Engineering, Universitas Gadjah Mada (UGM), Indonesia, in 2015. His research interests are intelligent systems, recommender agent, and information technology.

Chih-Hsien Hsia received the Ph.D. degree in Electrical Engineering from Tamkang University, New Taipei, Taiwan, in 2010. He was an Associate Professor with the Chinese Culture University and National Ilan University from 2015 to 2017. He currently is a Professor with the Department of Computer Science and Information Engineering, National Ilan University, Taiwan. Dr. Hsia is the Chapter Chair of IEEE Young Professionals Group, Taipei Section, and the Director of the IET Taipei Local Network. He serves as an Associate Editor of the Journal of Imaging Science and Technology and the Journal of Computers. His research interests include DSP IC Design, Multimedia Signal Processing, and Cognitive Learning.