Image Denoising with Two-Dimensional Adaptive Filter Algorithms

M. Shams Esfand Abadi* and S. Nikbakht*

Abstract: Two-dimensional (2D) adaptive filtering is a technique that can be applied to many image and signal processing applications. This paper extends the one-dimensional adaptive filter algorithms to 2D structure and the novel 2D adaptive filters are established. Based on this extension, the 2D variable step-size normalized least mean squares (2D-VSS-NLMS), the 2D-VSS affine projection algorithms (2D-VSS-APA), the 2D set-membership NLMS (2D-SM-NLMS), the 2D-SM-APA, the 2D selective partial update NLMS (2D-SPU-NLMS), and the 2D-SPU-APA are presented. In 2D-VSS adaptive filters, the step-size changes during the adaptation which leads to improve the performance of the algorithms. In 2D-SM adaptive filter algorithms, the filter coefficients are not updated at each iteration. Therefore, the computational complexity is reduced. In 2D-SPU adaptive algorithms, the filter coefficients are partially updated which reduce the computational complexity. We demonstrate the good performance of the proposed algorithms thorough several simulation results in 2D adaptive noise cancellation (2D-ANC) for image denoising. The results are compared with the classical 2D adaptive filters such as 2D-LMS, 2D-NLMS, and 2D-APA.

Keywords: Two Dimensional, Adaptive Filter, Noise Cancellation, Selective Partial Update, Variable Step-Size, Set-Membership.

1 Introduction

Adaptive filter algorithms have numerous applications in electrical engineering [1], [2] and [3]. Two-dimensional (2D) adaptive filters as well as one dimensional adaptive filter have received a great deal of attention in the last two decades [4], and that is because of their ability to take into account the inherent nonstationary statistical properties of two dimensional data, as well as 2D statistical correlation. The 2D adaptive filters have been applied to a variety of image processing applications such as image denoising, image enhancement, adaptive noise cancellation, 2D adaptive line enhancer, and 2D system identification. In [5], the one dimensional least mean squares (LMS) adaptive algorithm was extended to the 2D application and this algorithm was used for estimation of nonstationary images. In [6] an algorithm was proposed which was used the McClellan transformation. The new 2D-LMS whose convergence properties are not restricted to the one direction, was proposed in [7]. Also, the development of a 2D adaptive filter using the block diagonal LMS method was presented in [8].

The 2D-LMS adaptive filter [5] is essentially an extension of its one dimensional counterpart. The 2D-LMS is an attractive adaptation algorithm because of its simple structure, but this algorithm is highly sensitive to eigenvalue disparity, and its convergence speed is slow that is not appropriate in many applications. Therefore, to overcome this problem, the 2D normalized NLMS (2D-NLMS) algorithm was proposed. In this algorithm, the influence of the magnitude of the filter input on the convergence speed was considered. The 2D adaptive FIR filters which was based on affine projection algorithm (APA) was firstly introduced in [4]. In this algorithm the positions of projection vectors can be selected freely, and the performance is improved especially when the input data is highly correlated. Unfortunately, this improvement comes at the expense of a higher computational complexity. In [9] a fast APA for two dimensional adaptive linear filtering was presented. The results show that this algorithm has a fast convergence speed and good tracking ability. The 2D recursive least squares (2D-RLS) algorithm was proposed in [10-12]. Whereas the computational

Iranian Journal of Electrical & Electronic Engineering, 2011.
Paper first received 14 Nov. 2010 and in revised form 14 Mar. 2011.
* The Authors are with the Faculty of Electrical and Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran.
E-mails: mshams@srttu.edu, sr_nikbakht@yahoo.com.

complexity of one dimensional RLS is high, when we extend the one dimensional to 2D, the computational complexity is increased. The 2D-RLS has good performance in many applications, but the cost that we have to pay to enjoy its abilities is so expensive, therefore we did not consider this algorithm.

In 2D adaptive filter algorithms, the small variation of the step-sizes can produce an undesirably large change in adaptation speed and accuracy. Hence the optimal step-size selection is important in different applications. This selection is usually obtained by trial and error. Furthermore, an adaptive system with a constant step-size cannot appropriately adjust its parameters. To overcome this problem, the time-varying step-size technique was proposed in [13]. In [14], the variable step-size APA (VSS-APA), and variable step-size NLMS (VSS-NLMS) algorithm for one dimensional case were presented. The same approach in [14] was successfully extended to the other adaptive filter algorithms in [15] and [16]. In this paper, with the purpose of using variable step size in 2D applications, we extend the approach in [14] to establish of two new 2D adaptive filter algorithms which are called 2D-VSS-APA, and 2D-VSS-NLMS algorithms. In simulation results section, we demonstrate the good performance of the proposed algorithms in adaptive noise cancellation in digital images for image denoising. Unfortunately, when we use time varying step-size, we have to pay its cost, because of increasing the computational complexity.

Another way to overcome, the problem of existence tradeoff between low misadjustment and high convergence speed contemporaneous, is using the concept of set-membership (SM) filtering. In this method, by definition an upper bound on the estimation error, the number of adaptation of filter coefficients is reduced. The one dimensional SM-NLMS algorithm and the SM-APA were proposed in [17] and [18], respectively. To reduce the computational complexity in 2D applications, we introduced two new 2D-SM adaptive algorithms which are an extension of their one dimensional counterpart. The simulation results of the 2D-SM-NLMS and 2D-SM-APA show that these algorithms have good performance in elimination of noise in digital images.

In the classical adaptive filters the filter coefficients are fully updated. To reduce the computational complexity, other adaptive filter algorithms were introduced where the filter coefficients are partially updated. Based on this approach the filter coefficients which should be updated are optimally selected during the adaptation [19-24]. The one dimensional selective partial update NLMS (SPU-NLMS) and SPU-APA are important examples of these adaptive filters [25-26]. To reduce the computational complexity of conventional 2D-NLMS and 2D-APA algorithms, we extend the SPU approach to 2D structure to establish of the 2D-SPU-NLMS and 2D-SPU-APA.

The parameters selection of many 2D adaptive filter algorithms did not considered completely in the literatures. Many of the parameters have been selected by try and error approach in different literatures. In this paper we study the former and new 2D-algorithms comprehensively. As we know, each algorithm has different behavior in various applications of adaptive filters. So, we consider the performance of the presented algorithms in 2D adaptive noise cancellation (2D-ANC) for image denoising.

What we propose in this paper can be summarized as follows:

1. Extension of VSS approach to 2D-NLMS, and 2D-APA, and establishment of 2D-VSS-NLMS, and 2D-VSS-APA.
2. Extension of SPU approach to 2D-NLMS, and 2D-APA, and establishment of 2D-SPU-NLMS, and 2D-SPU-APA.
3. Extension of SM filtering to 2D-NLMS, and 2D-APA, and establishment of 2D-SM-NLMS, and 2D-SM-APA.
4. Demonstration of the presented algorithms in 2D-ANC application.

We have organized our paper as follows. Section 2 presents classical 2D adaptive filter algorithms. In section 3, the novel 2D adaptive filter algorithms are established. Section 4 presents the computational complexity of the derived algorithms. We conclude the paper by presenting several simulation results in 2D adaptive noise cancellation for reduction of noise in digital images.

Throughout the paper the following notations are adopted:

$(.)^T$	Transpose of vector or a matrix
$\mathrm{diag}(.)$	Diagonal of a matrix
$\mathrm{Tr}(.)$	Trace of a matrix
$\lVert . \rVert^2$	Squared Euclidean norm of a vector
$\lvert . \rvert$	Absolute value of a scalar
$E[.]$	Expectation operator

2 Background on Classical 2D Adaptive Filter Algorithms

As we know, linear system parameterization is an important class of system modeling with a wide area of applications. The most popular among the class of linear model is the finite impulse response (FIR). It is imposed in order to simplify the estimation task and to reduce the computational load in real-time application [9]. Let $u(i, j)$ be the input of a linear 2D FIR model, defined over a regularly spaced lattice $(i,j) \in [M_1, M_2]$, where M_1 and M_2 specify the order of the input data. The output of the 2D finite impulse response (FIR) digital filter, $y(i,j)$, is given by 2D finite impulse response (FIR) digital filter, $y(i,j)$, is given by

$$y(i,j) = \sum_{t=0}^{N_1-1} \sum_{l=0}^{N_2-1} w(t,l)\, u(i-t, j-l) \tag{1}$$

where, u(i,j) is the input signal, $w(t,l)$ is the model coefficients, and N_1 and N_2 specify the order of the FIR filter. Usually, the 2D signal is presented as a matrix. Therefore, the weight matrix $\mathbf{W}(i,j)$ and the input matrix $\mathbf{U}(i,j)$ are introduced as

$$\mathbf{W}_k(i,j) = \begin{bmatrix} w(0,0) & \dots & w(0,N_2-1) \\ \dots & \ddots & \dots \\ w(N_1-1,0) & \dots & w(N_1-1,N_2-1) \end{bmatrix} \tag{2}$$

$$\mathbf{U}_k(i,j) = \begin{bmatrix} u(i,j) & \cdots & u(i,j-N_2+1) \\ \dots & \ddots & \cdots \\ u(i-N_1+1,j) & \dots & u(i-N_1+1,j-N_2+1) \end{bmatrix} \tag{3}$$

where k is the iteration number and $0 \le k \le M_1 M_2$. Hadhoud expressed in [5] that the weight matrix and the input matrix can be converted into their one-dimensional form by lexicographic ordering. Equations (4) and (5) present the one dimensional form of Eq. (2), and Eq. (3).

$$\begin{aligned} \mathbf{w}_k(i,j) = [&w(0,0)\, w(0,1) \dots w(0,N_2-1) \\ &w(1,0) \dots w(N_1-1,N_2-1)]^T \end{aligned} \tag{4}$$

$$\begin{aligned} \mathbf{u}_k(i,j) = [&u(i,j)\, u(i,j-1) \dots u(i,j-N_2+1) \\ &u(i-1,j) \dots u(i-N_1+1, j-N_2+1)]^T \end{aligned} \tag{5}$$

Both vectors $\mathbf{u}(i,j)$ and $\mathbf{w}(i,j)$ have dimensions $(N_1 N_2) \times 1$. From Eq. (4), and Eq. (5), Eq. (1) can be stated as

$$y_k(i,j) = \mathbf{w}_k^T(i,j)\, \mathbf{u}_k(i,j) \tag{6}$$

2.1 2D-LMS Adaptive Filter Algorithm

This algorithm is based on the steepest descent method [27-29], and in this method the two dimensional weight adaptation is given by

$$\mathbf{w}_{k+1}(i,j) = \mathbf{w}_k(i,j) - \mu \frac{\delta \xi_k(i,j)}{\delta \mathbf{w}_k(i,j)} \tag{7}$$

where μ is the step size and can control the rate of convergence, steady state error, and filter stability and ξ, mean square error (MSE), is the cost function which is defined as $\xi_k(i,j) = E\left[e_k^2(i,j)\right]$, where $e_k(i,j)$ is the error signal at the k^{th} iteration and is given by

$$e_k(i,j) = d_k(i,j) - \mathbf{w}_k^T(i,j)\, \mathbf{u}_k(i,j) \tag{8}$$

where $d_k(i,j)$ is the desired signal. The aim of 2D-LMS algorithm is to obtained the optimum weight matrix such that the cost function, $\xi_k(i,j)$, is minimized. The 2D-LMS algorithm is a practical scheme for realizing 2D wiener filter, without explicitly solving the Wiener-Hopf equation. This algorithm uses the instantaneous estimates into the steepest descent algorithm. This algorithm replaces the cost function, $\xi_k(i,j) = E\left[e_k^2(i,j)\right]$ with $\xi_k(i,j) = \left[e_k^2(i,j)\right]$ which leads to

$$\frac{\delta \xi_k(i,j)}{\delta \mathbf{w}_k(i,j)} = \frac{2\, e_k(i,j)\, \delta(e_k(i,j))}{\delta \mathbf{w}_k(i,j)} = -2 e_k(i,j)\, \mathbf{u}_k(i,j) \tag{9}$$

By substituting Eq. (9) into Eq. (7), the filter coefficients update equation for 2D-LMS is obtained by

$$\mathbf{w}_{k+1}(i,j) = \mathbf{w}_k(i,j) + 2\mu\, e_k(i,j)\, \mathbf{u}_k(i,j) \tag{10}$$

2.2 2D-NLMS Adaptive Filter Algorithm

In this algorithm, as we mentioned before, the influence of magnitude of the filter input was considered. The filter coefficients update equation for 2D-NLMS algorithm is obtained by

$$\begin{aligned} \mathbf{w}_{k+1}(i,j) = \mathbf{w}_k(i,j) &+ \mu\, \mathbf{u}_k(i,j) \\ &(\mathbf{u}_k^T(i,j)\, \mathbf{u}_k(i,j) + \delta)^{-1}\, e_k(i,j) \end{aligned} \tag{11}$$

where δ is a positive small parameter which keep $\mathbf{u}_k(i,j)$ to become singular. We can consider this algorithm as a 2D adaptive filter algorithm with time varying step-size, when we define the variable step-size as

$$\mu = (\mathbf{u}_k^T(i,j)\, \mathbf{u}_k(i,j) + \delta)^{-1} \tag{12}$$

2.3 2D-APA Adaptive Filter Algorithm

The 2D-APA algorithm of Muneyasu and Hinamoto [4] can be interpreted as the 2D filter that minimizes the following objective function,

$$\min \left\| \mathbf{w}_{k+1}(i,j) - \mathbf{w}_k(i,j) \right\|^2 \tag{13}$$

subject to

$$\mathbf{d}_k(i,j) = \mathbf{w}_{k+1}^T(i,j)\tilde{\mathbf{U}}_k(i,j) \tag{14}$$

In 2D-APA, we use (KL) blocks to update the weight coefficients, where the parameters K and L are introduced to apply the recent input vectors in relations, and they are usually selected as $0 \le K \le N_1$, and $0 \le L \le N_2$. The matrix $\tilde{\mathbf{U}}(i,j)$ have dimension $(N_1N_2)\times(KL)$ and it consist of regressor vectors that belongs to the affine projection support region. By defining (15) for $m = 0,1,...K-1$

$$\hat{\mathbf{U}}_k(i-m,j) = [\mathbf{u}_k(i-m,j)\mathbf{u}_k(i-m,j-1) \\ ...\mathbf{u}_k(i-m,j-L+1)] \tag{15}$$

The input matrix can be obtained as

$$\tilde{\mathbf{U}}_k(i,j) = [\hat{\mathbf{U}}_k(i,j)\hat{\mathbf{U}}_k(i-1,j)...\hat{\mathbf{U}}_k(i-K+1,j)] \tag{16}$$

Also, $\mathbf{d}_k(i,j)$ is the desired signal vector with dimension of $(KL)\times 1$ which is given by

$$\mathbf{d}_k(i,j) = [d(i,j)d(i,j-1)...d(i,j-L+1) \\ d(i-1,j)...d(i-K+1,j-L+1)]^T \tag{17}$$

From the above, the filter coefficients update equation for 2D-APA can be established by

$$\mathbf{w}_{k+1}(i,j) = \mathbf{w}_k(i,j) + \mu\,\Delta\mathbf{w}_k(i,j) \tag{18}$$

where

$$\Delta\mathbf{w}_k(i,j) = \tilde{\mathbf{U}}_k(i,j)(\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j)+\delta\mathbf{I})^{-1}\mathbf{e}_k(i,j) \tag{19}$$

and

$$\mathbf{e}_k(i,j) = \mathbf{d}_k(i,j) - \tilde{\mathbf{U}}_k^T(i,j)\mathbf{w}_k(i,j) \tag{20}$$

is the output error vector. Notice that the 2D-NLMS algorithm is a special case of 2D-APA algorithm, when we use a current block ($K=0, L=0$) to update the filter coefficients.

3 Derivation of 2D Adaptive Filter Algorithms

In this section we introduce the novel 2D adaptive filter algorithms.

3.1 2D-VSS-APA and 2D-VSS-NLMS

In Eq. (18) the weight update equation for 2D-APA was presented. Following the same approach in [14], if we rewrite the update equation for 2D-APA respect to weight error vector, $\tilde{\mathbf{w}}_k(i,j) = \mathbf{w}^{\mathbf{o}}(i,j) - \mathbf{w}_k(i,j)$, where $\mathbf{w}^{\mathbf{o}}(i,j)$ is the true unknown filter vector, Eq. (21) is obtained as

$$\tilde{\mathbf{w}}_{k+1}(i,j) = \tilde{\mathbf{w}}_k(i,j) - \mu\tilde{\mathbf{U}}_k(i,j)(\tilde{\mathbf{U}}_k^T(i,j) \\ \tilde{\mathbf{U}}_k(i,j)+\delta\mathbf{I})^{-1}\mathbf{e}_k(i,j) \tag{21}$$

By taking the squared Euclidean norm and expectations from both sides of Eq. (21), Eq. (22) is obtained as

$$E\|\tilde{\mathbf{w}}_{k+1}(i,j)\|^2 = E\|\tilde{\mathbf{w}}_k(i,j)\|^2 - \Delta\mu \tag{22}$$

where

$$\Delta\mu = \mu E\left[\mathbf{e}_k^T(i,j)(\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j))^{-1}\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{w}}_k(i,j)\right] + \\ \mu E\left[\tilde{\mathbf{w}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j)(\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j))^{-1}\mathbf{e}_k(i,j)\right] - \\ \mu^2 E\left[\mathbf{e}_k^T(i,j)(\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j))^{-1}\mathbf{e}_k(i,j)\right] \tag{23}$$

To have maximum decreasing in MSD from iteration (k) to $(k+1)$, $\Delta\mu$ should be maximized. Maximizing $\Delta\mu$ with respect to μ, the optimum step-size can be stated as

$$\mu_k^{\circ}(i,j) = \frac{\operatorname{Re} E\left[\mathbf{e}_k^T(i,j)(\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j))^{-1}\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{w}}_k(i,j)\right]}{E\left[\mathbf{e}_k^T(i,j)(\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j))^{-1}\mathbf{e}_k(i,j)\right]} \tag{24}$$

If we define the linear data model for desired signal as $\mathbf{d}_k(i,j) = \tilde{\mathbf{U}}_k^T(i,j)\mathbf{w}^{\mathbf{o}}(i,j) + \mathbf{v}_k(i,j)$, where $\mathbf{v}_k(i,j)$ is the vector of measurement noise, the error vector can be described as

$$\mathbf{e}_k(i,j) = \tilde{\mathbf{U}}_k^T(i,j)\mathbf{w}_k^{\circ}(i,j) + \mathbf{v}_k(i,j) - \tilde{\mathbf{U}}_k^T(i,j)\mathbf{w}_k(i,j) \\ = \tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{w}}_k(i,j) + \mathbf{v}_k(i,j) \tag{25}$$

We assume that $\mathbf{v}_k(i,j)$ is statistically independent of regression data $\tilde{\mathbf{U}}_k(i,j)$. Substituting Eq. (25) into Eq. (24), Eq. (26) is given by

$$\mu_k^{\circ}(i.j) \approx E[(\tilde{\mathbf{w}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j) + \mathbf{v}_k(i,j))(\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j))^{-1} \\ \tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{w}}_k(i,j)]/E[(\tilde{\mathbf{w}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j) + \mathbf{v}_k^T(i,j)) \\ (\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j))^{-1}(\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{w}}_k(i,j) + \mathbf{v}_k(i,j))] \tag{26}$$

By defining:

$$\mathbf{q}_k(i,j) = \tilde{\mathbf{U}}_k(i,j)(\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j))^{-1} \\ \tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{w}}_k(i,j) \tag{27}$$

where $\mathbf{q}_k(i,j)$ is $(N_1N_2)\times 1$ column vector, we obtain

$$\|\mathbf{q}_k(i,j)\|^2 = \tilde{\mathbf{w}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j)(\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j))^{-1} \\ \tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{w}}_k(i,j) \tag{28}$$

Therefore, Eq. (26) can be written as

$$\mu_k^{\circ}(i.j)=\frac{E\|\mathbf{q}_k(i,j)\|^2}{E\|\mathbf{q}_k(i,j)\|^2+\sigma_v^2\mathrm{Tr}\left\{E\left((\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j))^{-1}\right)\right\}} \quad (29)$$

where σ_v^2 is the variance of measurement noise.

To obtain the optimum step-size from Eq. (29), we need $\mathbf{w^o}(i,j)$, which is unknown. From Eq. (25), we can estimate $\mathbf{q}_k(i,j)$ by time averaging as follows:

$$\begin{aligned}\hat{\mathbf{q}}_{k+1}(i,j)&=\gamma\hat{\mathbf{q}}_k(i,j)+(1-\gamma)\tilde{\mathbf{U}}_k(i,j)\\&(\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j)+\delta\mathbf{I})^{-1}\mathbf{e}_k(i,j)\end{aligned} \quad (30)$$

where γ is a smoothing factor ($0\le\gamma<1$).

By substituting the estimated value of $\mathbf{q}_k(i,j)$ instead of its real value in Eq. (29), the variable step-size for 2D-APA algorithm is given by

$$\mu_k(i,j)=\mu_{max}\frac{\|\hat{\mathbf{q}}_k(i,j)\|^2}{\|\hat{\mathbf{q}}_k(i,j)\|^2+C} \quad (31)$$

where $C=\sigma_v^2\mathrm{Tr}\left\{E\left((\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j))^{-1}\right)\right\}$ and can be selected constant in simulation results. To guarantee the stability of the filter μ_{max} is chosen less than 2. Therefore, the filter coefficients update equation for 2D-VSS-APA algorithm can be stated as

$$\begin{aligned}\mathbf{w}_{k+1}(i,j)&=\mathbf{w}_k(i,j)+\mu_k(i,j)\tilde{\mathbf{U}}_k(i,j)\\&(\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j)+\delta\mathbf{I})^{-1}\mathbf{e}_k(i,j)\end{aligned} \quad (32)$$

2D-VSS-NLMS is a special case of 2D-VSS-APA when we use only one block to update weight matrix. The filter coefficients update equation in 2D-VSS-NLMS is given by the following relations

$$\begin{aligned}\mathbf{w}_{k+1}(i,j)&=\mathbf{w}_k(i,j)+\mu_{max}\frac{\|\hat{\mathbf{q}}_k(i,j)\|^2}{\|\hat{\mathbf{q}}_k(i,j)\|^2+C}\\&\mathbf{u}_k(i,j)(\mathbf{u}_k^T(i,j)\mathbf{u}_k(i,j)+\delta)^{-1}e_k(i,j)\end{aligned} \quad (33)$$

where

$$\begin{aligned}\hat{\mathbf{q}}_{k+1}(i,j)&=\gamma\hat{\mathbf{q}}_k(i,j)+(1-\gamma)\mathbf{u}_k(i,j)\\&(\mathbf{u}_k^T(i,j)\mathbf{u}_k(i,j)+\delta)^{-1}e_k(i,j)\end{aligned} \quad (34)$$

3.2 2D-SM-APA and 2D-SM-NLMS

In 2D set-membership filtering, an upper bound, β, on the magnitude of the estimation error is specified. The parameter β can vary with the specific application. When the signal error is larger than the certain value (β), the filter coefficients are updated. On the other word, the step-size which is proportionate to the absolute value of error is introduced in this algorithm. Following the same approach in [18] for one dimensional SM-APA, the weight update equation for 2D-SM-APA is given by

$$\begin{aligned}\mathbf{w}_{k+1}(i,j)&=\mathbf{w}_k(i,j)+\alpha_k(i,j)\tilde{\mathbf{U}}_k(i,j)\\&(\tilde{\mathbf{U}}_k^T(i,j)\tilde{\mathbf{U}}_k(i,j)+\delta\mathbf{I})^{-1}\mathbf{e}_k(i,j)\,\mathbf{v}_1\end{aligned} \quad (35)$$

where

$$\alpha_k(i,j)=\begin{cases}1-\dfrac{\beta}{|e_k(i,j)|} & \text{if } |e_k(i,j)|>\beta\\ 0 & \text{otherwise}\end{cases} \quad (36)$$

and

$$\begin{aligned}\mathbf{e}_k(i,j)=[&e_k(i,j)e_k(i,j-1)...e_k(i,j-L+1)\\&e_k(i-1,j)...e_k(i-K+1,j-L+1)]^T\end{aligned} \quad (37)$$

$$\mathbf{v}_1=[1\ 0\ 0....0]^T$$

The 2D-SM-NLMS is a special case of 2D-SM-APA and is obtained when we use current block to update the weight matrix and can be stated as

$$\begin{aligned}\mathbf{w}_{k+1}(i,j)&=\mathbf{w}_k(i,j)+\alpha_k(i,j)\mathbf{u}_k(i,j)\\&(\mathbf{u}_k^T(i,j)\mathbf{u}_k(i,j)+\delta)^{-1}e_k(i,j)\end{aligned} \quad (38)$$

It is clear in the above equation that, if the absolute value of error becomes smaller than β, then the step-size will be zero and if it is larger than β, the step-size becomes large (close to 1). These algorithms exhibit like VSS adaptive filters and also reduce the computational complexity.

3.3 2D-SPU-NLMS

As we mentioned, we introduce 2D-SPU-NLMS algorithm to reduce the computational complexity. In this algorithm, we partitioned the input matrix into N_1 blocks, each of length N_2, and in each iteration a subset of these blocks is updated. The parameter S is used to show the number of blocks to update in each iteration. By extending the approach in [20] and [25] to 2D version, the filter coefficients update equation for 2D-SPU-NLMS is obtained by

$$\mathbf{w}_{k+1}(i,j)=\mathbf{w}_k(i,j)+\mu\mathbf{C}_k(i,j)\mathbf{u}_k(i,j)e_k(i,j) \quad (39)$$

where

$$\mathbf{C}_k(i,j)=\frac{\mathbf{A}_k}{\|\mathbf{A}_k\,\mathbf{u}_k(i,j)\|^2} \quad (40)$$

The matrix $\mathbf{A}_k$ is a $(N_1N_2)\times(N_1N_2)$ diagonal matrix with the **1** and **0** blocks each of length N_2 on the diagonal and the positions of 1's on the diagonal determine which coefficients should be updated in each iteration.

By partitioning the regressor vector $\mathbf{u}_k(i,j)$ into N_1 blocks each of length N_2, the positions of **1** blocks (S blocks and $S \le N_1$) on the diagonal of $\mathbf{A}_k$ matrix for each iteration in the 2D-SPU-NLMS adaptive algorithms are determined by the following procedure:

1. The $\|\hat{\mathbf{u}}_{kk'}(i,j)\|^2$ values are sorted for $0 \le k' \le (N_1 - 1)$, where $\hat{\mathbf{u}}_{kk'}(i,j)$ describes each block of input matrix at k^{th} iteration, and $\hat{\mathbf{u}}_{kk'}(i,j) = [u(i-k',j)\, u(i-k',j-1)\, u(i-k',j-2)$ $.....u(i-k',j-N_2+1)]^T$.
2. The k' values that determine the positions of 1 blocks correspond to the S largest values of $\|\hat{\mathbf{u}}_{kk'}(i,j)\|^2$.

3.4 2D-SPU-APA

The SPU approach can be extended to APA. As we mentioned, the SPU-APA for one dimensional was presented [20] and [26]. This section presents the 2D-SPU-APA to reduce the computational complexity in two dimensional applications, where we use (KL) blocks to update the weights matrix. In this algorithm we partition the input matrix into N_1 blocks, each of length $(N_2 \times KL)$, and we use S blocks to update. The filter coefficients update equation for 2D-SPU-APA is given by

$$\mathbf{w}_{k+1}(i,j) = \mathbf{w}_k(i,j) + \mu\, \mathbf{A}_k \tilde{\mathbf{U}}_k(i,j) \\ (\tilde{\mathbf{U}}_k^T(i,j)\, \mathbf{A}_k \tilde{\mathbf{U}}_k(i,j) + \delta \mathbf{I})^{-1} \mathbf{e}_k(i,j) \quad (41)$$

where the $\mathbf{A}_k$ matrix is the $(N_1N_2 \times N_1N_2)$ diagonal matrix with the **1** and **0** blocks each of length N_2 on the diagonal and the positions of 1's on the diagonal determine which coefficients should be updated in each iteration. By introducing $(N_2 \times KL)$ block matrix $\hat{\mathbf{U}}_{kk'}(i,j)$ for $0 \le k' \le (N_1 - 1)$ as

$$\hat{\mathbf{U}}_{kk'}(i,j) = [\hat{\mathbf{u}}_{kk'}(i,j)\, \hat{\mathbf{u}}_{kk'}(i,j-1) ... \hat{\mathbf{u}}_{kk'}(i,j-L+1) \\ \hat{\mathbf{u}}_{k(k'+1)}(i,j), ... \hat{\mathbf{u}}_{k(k'+1)}(i,j-L+1) ... \hat{\mathbf{u}}_{k(k'+K-1)}(i,j-L+1)]^T \quad (42)$$

the following procedure is used to find the positions of **1** blocks.

1. Compute $\mathrm{Tr}(\hat{\mathbf{U}}_{kk'}^T(i,j)\, \hat{\mathbf{U}}_{kk'}(i,j))$ for $0 \le k' \le (N_1 - 1)$
2. The k' values that determine the positions of 1 blocks correspond to the S largest values of $\mathrm{Tr}(\hat{\mathbf{U}}_{kk'}^T(i,j)\, \hat{\mathbf{U}}_{kk'}(i,j))$.

When all blocks are used for updating the weight matrix $(S = N_1)$, the conventional 2D-APA is established.

4 Computational Complexity

Table 1 presents the computational complexity of 2D adaptive algorithms which were introduced in this paper. As we can see, the computational complexity of 2D-SPU-NLMS and 2D-SPU-APA are less than conventional 2D-NLMS and 2D-APA algorithms, and the computational complexity for 2D-VSS-NLMS and 2D-VSS-APA algorithms are more than these algorithms. For the 2D-SPU adaptive filters, the number of comparisons based on heapsort algorithm have been also presented in Table 1 [30]. For the 2D-SM-NLMS, and 2D-SM-AP algorithms, the adaptation is related to the condition in Eq. (36). If the condition in Eq. (36) always becomes true (which in practice it does not), then the computational complexity of 2D-SM algorithms are similar to the complexity of classical 2D adaptive algorithms. But the gains of applying the 2D-SM algorithms comes through the reduced number of required updates, which cannot be accounted for a priori, and an increased performance as compared to classical 2D adaptive filter algorithms.

5 Simulation Results

In this section, we present the simulation results in one of the important applications of 2D adaptive filter algorithms, namely, 2D adaptive noise cancellation. Fig. 1 shows the setup of 2D adaptive noise cancellation. The primary signal is the combination of desired and noise signals and the reference signal is noise which is correlated with the noise in the primary signal.

$$e(i,j) = d(i,j) - v_2(i,j) \approx \hat{u}(i,j) \quad (43)$$

The 2D adaptive noise cancellation tries to eliminate the noise from the noisy signal. Based on Eq. (43), after the convergence of filter coefficients, e(i,j) will be the estimation of desired signal. In this simulation, the white Gaussian noise with zero mean and unit variance v(i,j) is added to the images to produce the noisy images where the signal to noise ratio (PSNR) is set to 0 dB. Based on Eq. (44), the reference signal, $v_1(i, j)$, is generated by passing the white Gaussian noise with zero mean and unit variance through the 2D low pass filter. In this section, we use four standard images. The dimension of each original image is 256×256. Figs. 2 and 3 show the original and noisy images. Also, the order of 2D adaptive filter is set to N1 = N2 = 5.

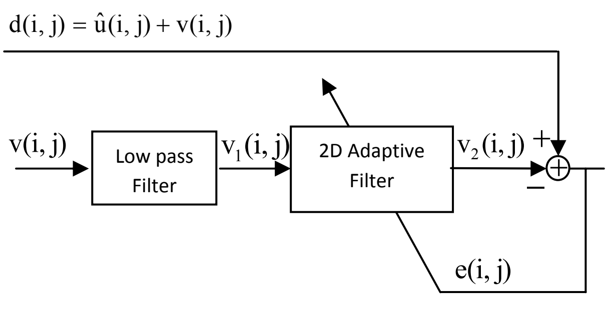


Fig. 1 Adaptive noise cancellation setup.

$$b(z_1) = 1 - 0.7z_1^{-1} + 0.5z_1^{-2} - 0.05z_1^{-3} + 0.0056z_1^{-4} - 0.0004z_1^{-5}$$
$$b(z_2) = 1 - 0.7z_2^{-1} + 0.5z_2^{-2} - 0.045z_2^{-3} + 0.0046z_2^{-4} - 0.0003z_2^{-5}$$
$$B(z_1, z_2) = b(z_1)b(z_2) \quad (44)$$
$$B(z_1, z_2)\, v_1(i, j) = v(i, j)$$

One of the important subjects that did not fully considered in the literature is the performance of 2D-algorithms in a comprehensive range of step-size. In Fig. 4, the performance of 2D-LMS, 2D-NLMS and 2D-APA algorithms in different values of step-size and for four images are considered. The step-size changes from 10^{-4} to 1. To compare the performance of 2D adaptive filters, we calculated the PSNR of the output image which is defined as

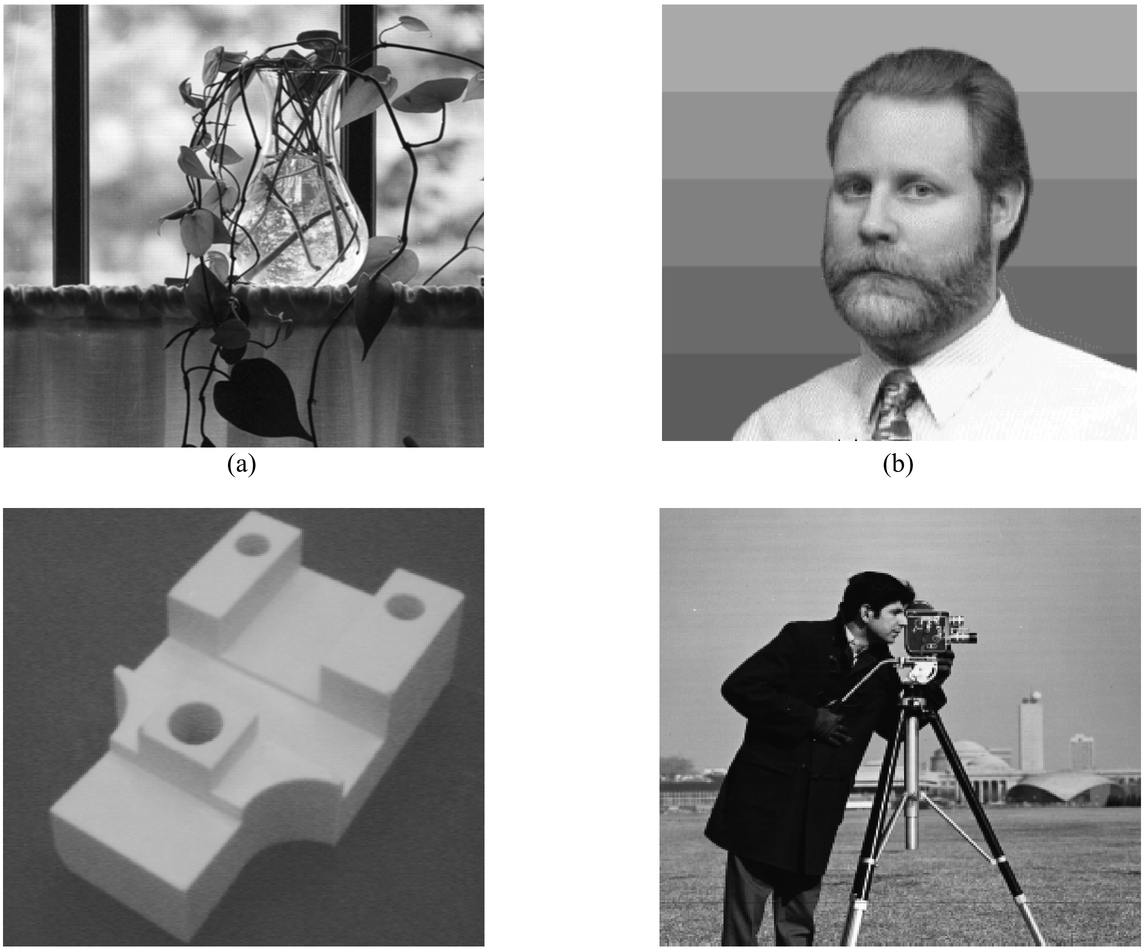
$$\mathrm{PSNR} = -10\log_{10}\left(\frac{\sum_{i=0}^{M_1-1}\sum_{j=0}^{M_2-1}\left(I(i,j) - J(i,j)\right)^2}{M_1 M_2}\right) \quad (45)$$

where I and J is the original and noisy images, respectively. Also, M_1 and M_2 describe the size of input images. Fig. 4 shows that for each algorithm, there is an optimum value for the step-size to have maximum PSNR in output image. This figure shows that the optimum step-size in 2D-LMS is 10^{-3} for four images. In 2D-NLMS, this value is approximately 10^{-1}, and for 2D-APA with K=2, and L=2, the optimum step-size is 10^{-2}. Also, the stability bound of 2D-LMS is less than 2D-NLMS, and 2D-APA algorithms.

Figure 5 shows the PSNR of output images versus the step-size for 2D-SPU-NLMS algorithm. Different values for S have been used in this simulation. As we can see, there is an optimum step-size for 2D-SPU-NLMS algorithm. Simulation results show that the optimum step-sizes are close to each other for different values of S. Also by increasing the parameter S, the PSNR of output image increases. This figure shows that the stability band of 2D-SPU-NLMS for S=1 and S=2 is less than 2D-SPU-NLMS with S=3, 4, and 5.

Table 1 The computational complexity of 2D-LMS, 2D-NLMS, 2D-APA, 2D-SM-NLMS, 2D-SM-APA, 2D-VSS-NLMS, 2D-VSS-APA, 2D-SPU-NLMS and 2D- SPU-APA.

Algorithm	Multiplications	Divisions	Additional Multiplications	Comparisons
2D-LMS	$2(N_1N_2)+1$	—	—	—
2D-NLMS	$3(N_1N_2)+1$	1	—	—
2D-APA	$(KL)^2(N_1N_2+1)+2(KL)N_1N_2+(KL)^3$	—	—	—
2D-SM-NLMS	$3(N_1N_2)+1$	2	—	—
2D-SM-APA	$(KL)^2(N_1N_2+1)+2(KL)N_1N_2+(KL)^3$	1	—	—
2D-VSS-NLMS	$3(N_1N_2)+1$	1	(N_1N_2)	—
2D-VSS-APA	$(KL)^2(N_1N_2+1)+2(KL)N_1N_2+(KL)^3$	1	(N_1N_2)	—
2D-SPU-NLMS	$3(SN_2)+1$	1	1	$N_1\log_2 S+O(N_1)$
2D-SPU-APA	$(KL)^2(SN_2+1)+2(KL)SN_2+(KL)^3$	—	1	$N_1\log_2 S+O(N_1)$



(a) (b) (c) (d)

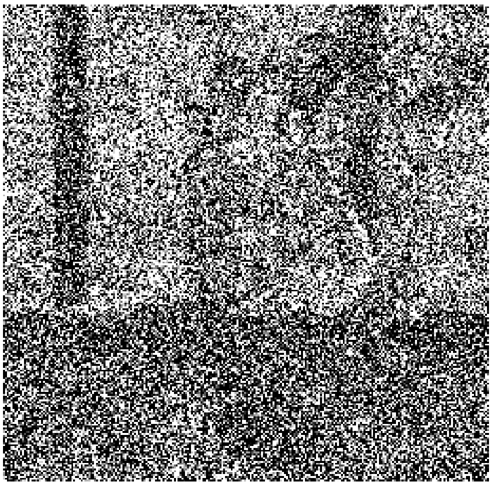
Fig. 2 Original images. (a) Pot (b) Simpson (c) Part (d) Camera man.

Figure 6 shows the PSNR of output images versus the step-size, for 2D-SPU-APA algorithm. Different values for S have been used in this simulation. As we can see, there is an optimum step-size for 2D-SPU-APA algorithm. Simulation results show that the optimum step-sizes are close to each other for different values of S. Also by increasing the parameter S, PSNR of output image increases. As we can see, the stability bound low values of **S** are less than large values.

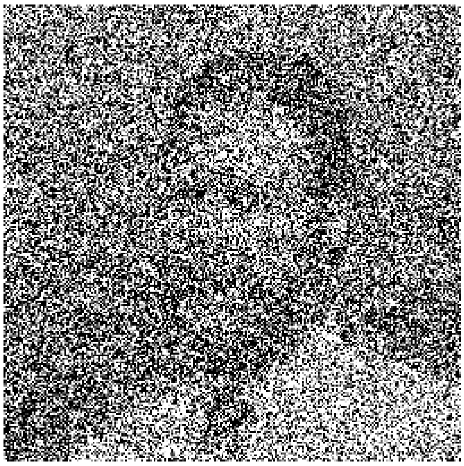
Table 2 shows the selected parameters and the PSNR of output image for different images and various 2D adaptive filter algorithms in optimum values. In this table we have also presented the results of applying 2D-VSS-NLMS, and 2D-VSS-APA. As we can see, the PSNR of output image in 2D-VSS-NLMS is better that 2D-NLMS adaptive filter algorithm. Also, the 2D-VSS-APA has better performance than 2D-APA.

Figure 7 shows the output images of 2D-LMS, 2D-NLMS, 2D-APA, 2D-VSS-NLMS, and 2D-VSS-APA. The results show that the 2D-VSS-APA has better results than other algorithms. In Figs. 8-10, we presented the results for different images. Again the same results can be seen in these figures.

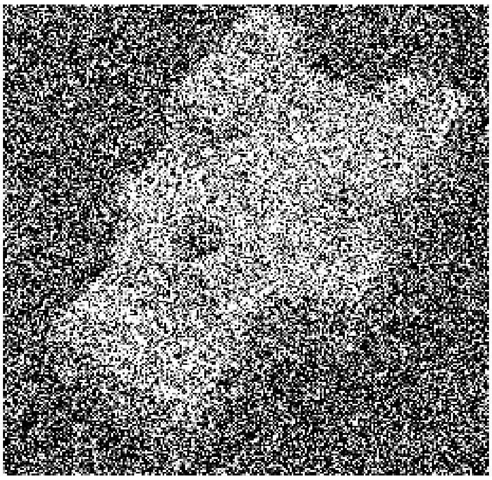
In the following, we applied the 2D-SM-NLMS, and 2D-SM-APA to denoising of images. Table 3 shows the selected parameters and the PSNR of output image for four images. This table also shows the number of filter coefficients in update for these algorithms. As we can see, the number of filter coefficients in update for 2D-SM-NLMS, and 2D-SM-APA is less than 2D-NLMS, and 2D-APA. Comparing the results from this table with Table 1 shows that the PSNR of output image based on 2D-SM adaptive filter algorithms is larger than ordinary 2D adaptive filters. Figs. 11 and 12 show the output images with 2D-SM-NLMS, and 2D-SM-APA respectively. The simulation results show that the 2D-SM adaptive filter algorithms have good ability to eliminate the noise from the noisy images.



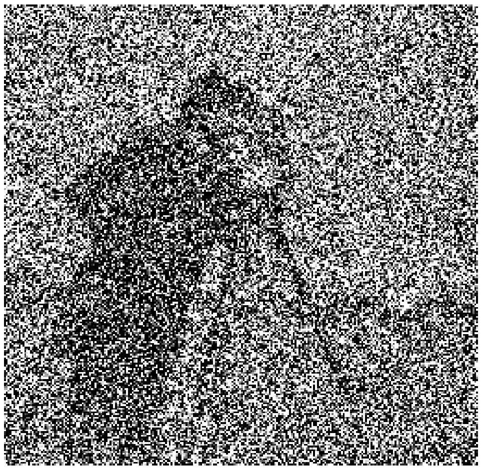
(a)



(b)



(c)



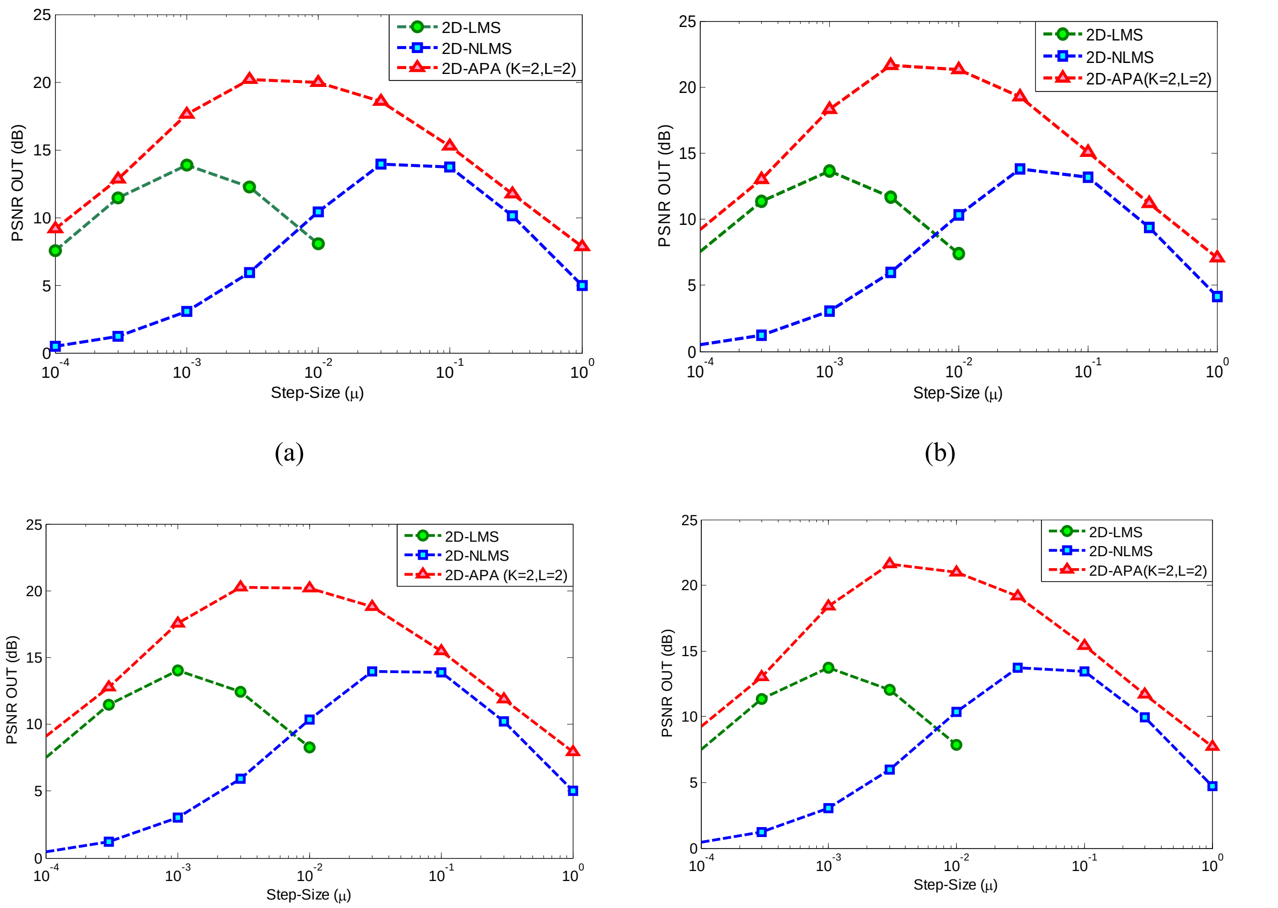
(d)

Fig. 3 Noisy images with PSNR=0. (a) Pot (b) Simpson (c) Part (d) Camera man.

Table 4 presents the results for 2D-SPU-NLMS, and 2D-SPU-APA. Different values for the parameter S have been used. As we can see by increasing the parameter S, the PSNR of output image increases. This fact can be seen for all images. Also, the results for S=3, 4, and 5 are very close together. Furthermore, the computational complexity of 2D-SPU adaptive filters is lower than ordinary algorithms.

In Table 5, we presented the executing time of the proposed algorithms in 2D-ANC for “Simpson” image. The processor characteristic of computer was Intel Core 2 Duo CPU 2.53 GHz with 4.00 GB RAM. The parameters of the algorithms are according to Tables 2, 3 and 4. It is clear that, the executing time of 2D-VSS algorithms are further than conventional algorithms. In 2D-SM-NLMS, the executing time is less than the classical 2D adaptive algorithms. In 2D-SPU adaptive algorithms, by increasing the parameter S, the executing time increases.

To complete our simulations, we justified the presented algorithms in different PSNR. Fig. 13 shows the output PSNR versus input PSNR for 2D-LMS, 2D-NLMS, 2DAPA, 2D-VSS-NLMS, and 2D-VSS-APA. The results show that 2D-VSS-APA has better performance than other algorithms. In Fig. 14, we presented the results for 2D-SPU-NLMS algorithms. As we can see, by increasing the parameter S, the output PSNR increases. Furthermore, the results for S=3, 4, and 5 are very close for different PSNR of input image. Figure 15 presents the results for 2D-SPU-APA. This figure shows that, the performance for S=2, 3, 4, and 5 are very close in various PSNR of input images.



(a) (b) (c) (d)

Fig. 4 Output PSNR of different images versus the step-size for 2D-LMS, 2D-NLMS, and 2D-APA with K=2, L=2.
(a) Pot (b) Simpson (c) Part (d) Camera man.

To complete our discussion about 2D adaptive algorithms, we consider the influence of the order of the filter, in 2D adaptive noise cancellation. In Figs. 16, 17 and 18, we justified the performance of 2D-LMS, 2D-NLMS and 2D-APA algorithms in different input PSNR and various orders of filter. In Fig. 16 the step-size was set to $\mu = 0.001$ and cameraman image was used. In Fig. 17, we considered $\mu = 0.01$ and Simpson image was used and finally in Fig. 18, $\mu = 0.01$ was set and the part image was used. The simulation results show that by increasing the order of the filter, the output PSNR decreases. Also the simulation results show that for large values of input PSNR, the PSNR improvement decreases.

Table 3 Comparison of PSNR Improvement for 2D-SM Adaptive Filters.

ALGORITHM	IMAGE	PARAMETER β	PSNR(OUT)	NUMBER OF WEIGHT UPDATE
2D-SM-NLMS	Pot	1	18.47	654
	Simpson	1	17.21	1582
	Part	1	18.21	1348
	Cameraman	1	18.25	752
2D-SM-APA (K=2, L=2)	Pot	1	20.21	751
	Simpson	1	20.39	1872
	Part	1	23.76	1597
	Cameraman	1	18.10	894

Table 4 Comparison of PSNR Improvement for 2D-SPU Adaptive Filters.

ALGORITHM	IMAGE	PARAMETER	PSNR(OUT) IN DIFFERENT NUMBER OF BLOCK				
		μ	S=5	S=4	S=3	S=2	S=1
2D-SPU-NLMS	Pot	0.05	14.47	14.32	14.14	13.60	13.22
	Simpson	0.05	14.22	14.09	13.88	13.31	12.67
	Part	0.05	14.26	14.11	13.94	13.37	12.99
	Cameraman	0.05	14.62	14.49	14.30	13.76	13.26
2D-SPU-APA	Pot	0.005	20.34	20.23	20.05	19.91	18.59
	Simpson	0.003	21.64	21.42	21.29	21.28	20
	Part	0.003	21.62	21.38	21.29	21.4	19.98
	Cameraman	0.003	20.25	20.09	20.01	20	19.16

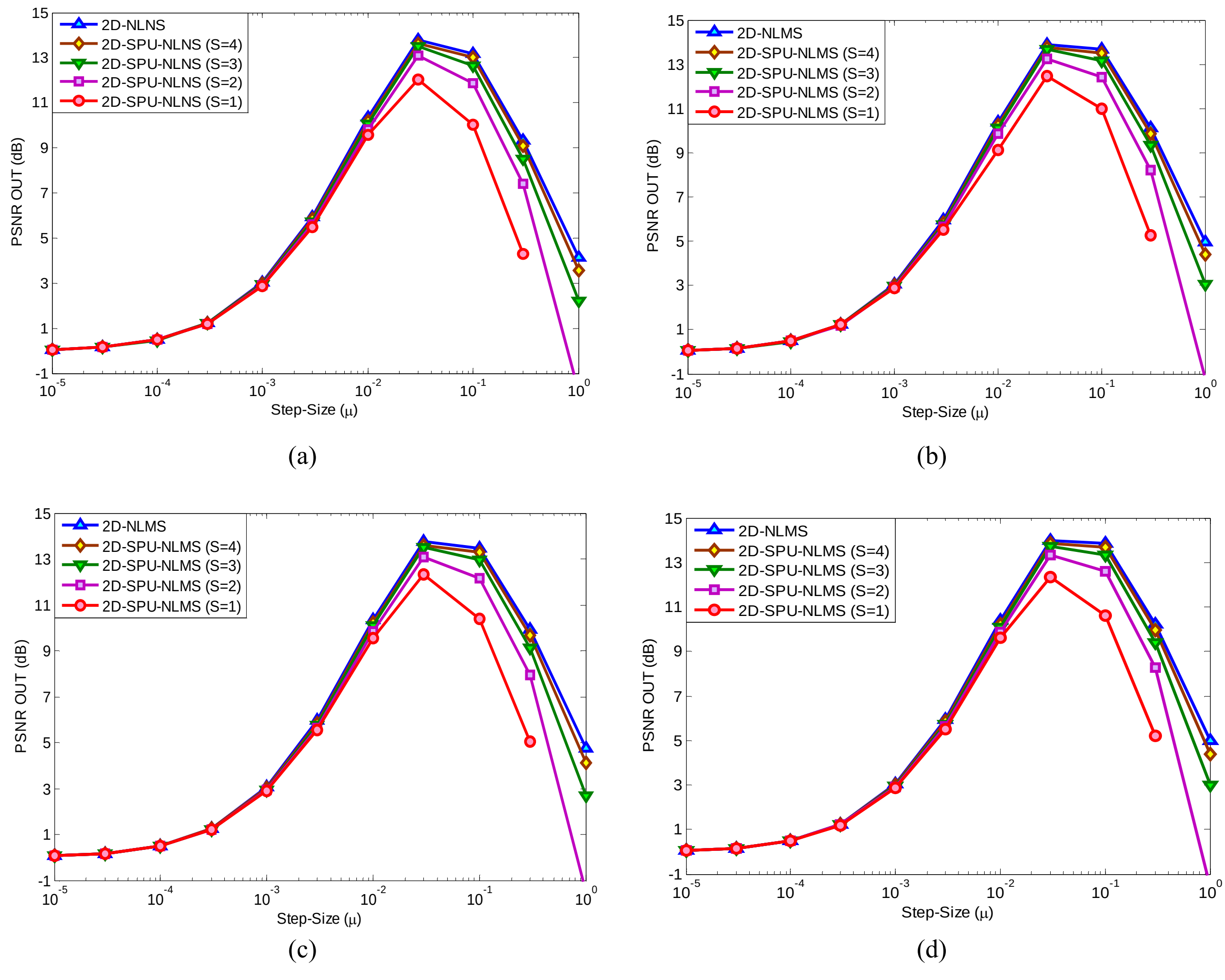


Fig. 5 Output PSNR of different images versus the step-size for 2D-SPU-NLMS with S=1, 2, 3, 4, and 5. (a) Pot (b) Simpson (c) Part (d) Camera man.

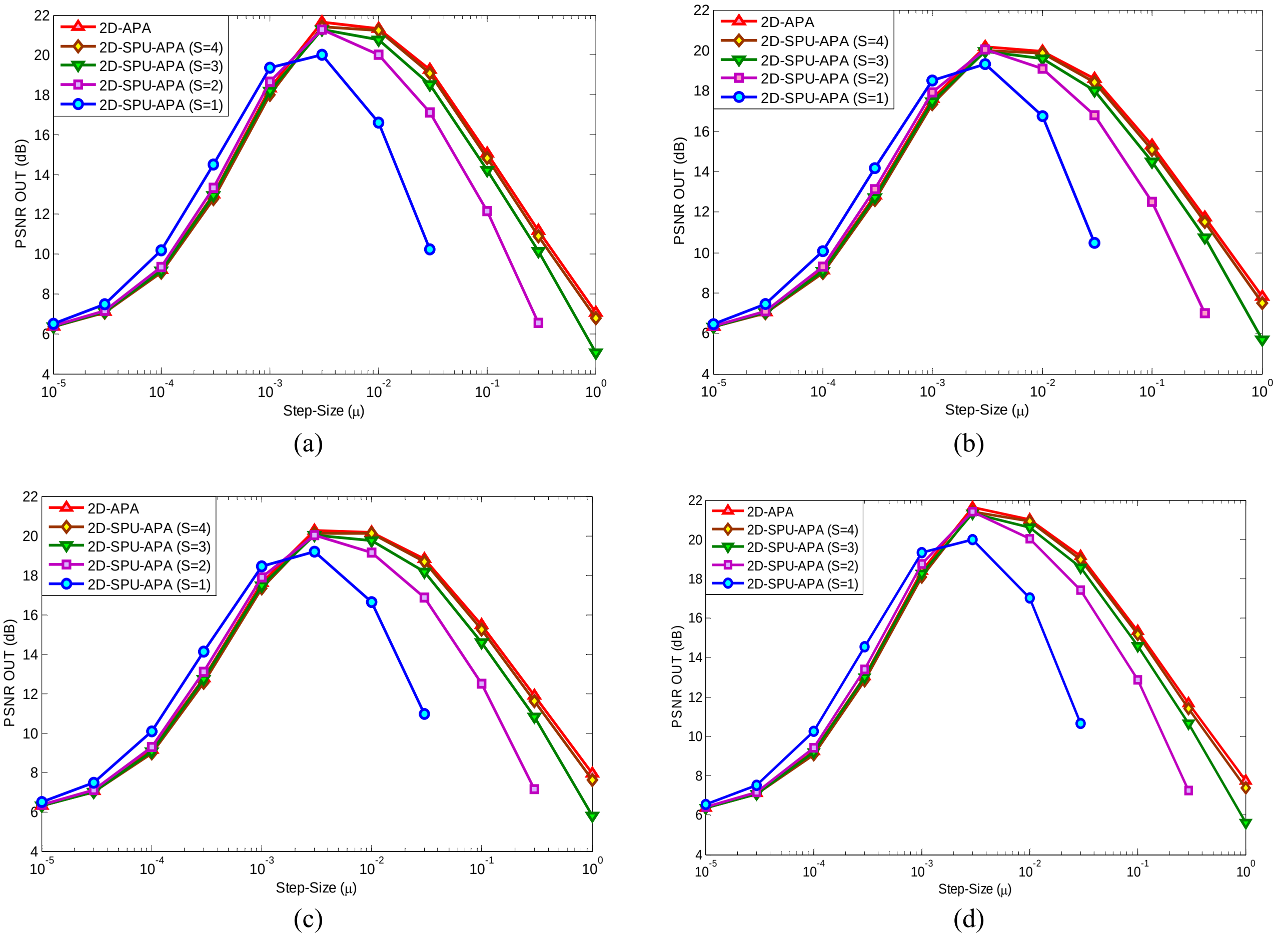


Fig. 6 Output PSNR of different images versus the step-size for 2D-SPU-APA with K=2, L=2 and various values for S. (a) Pot (b) Simpson (c) Part (d) Camera man.

Table 2 Comparison of PSNR Improvement for 2D Adaptive Filters.

ALGORITHM	IMAGE	PARAMETERS				PSNR(OUT)
		μ	C	γ	μ_{max}	
2D-LMS	Pot	0.001	------	------	------	13.74
	Simpson	0.001	------	------	------	13.64
	Part	0.001	------	------	------	13.69
	Cameraman	0.001	------	------	------	13.98
2D-NLMS	Pot	0.05	------	------	------	14.47
	Simpson	0.05	------	------	------	14.22
	Part	0.05	------	------	------	14.26
	Cameraman	0.05	------	------	------	14.62
2D-APA (K=2,L=2)	Pot	0.005	------	------	------	20.34
	Simpson	0.003	------	------	------	21.65
	Part	0.003	------	------	------	21.62
	Cameraman	0.003	------	------	------	20.25
2D-VSS-NLMS	Pot	------	0.0001	0.99	1	16.18
	Simpson	------	0.0002	0.99	1	16.34
	Part	------	0.0001	0.99	1	16.59
	Cameraman	------	0.0003	0.99	1	16.74
2D-VSS-APA (K=2,L=2)	Pot	------	0.01	0.99	1	21.87
	Simpson	------	0.01	0.99	1	23.35
	Part	------	0.01	0.99	1	23.59
	Cameraman	------	0.01	0.99	1	21.39

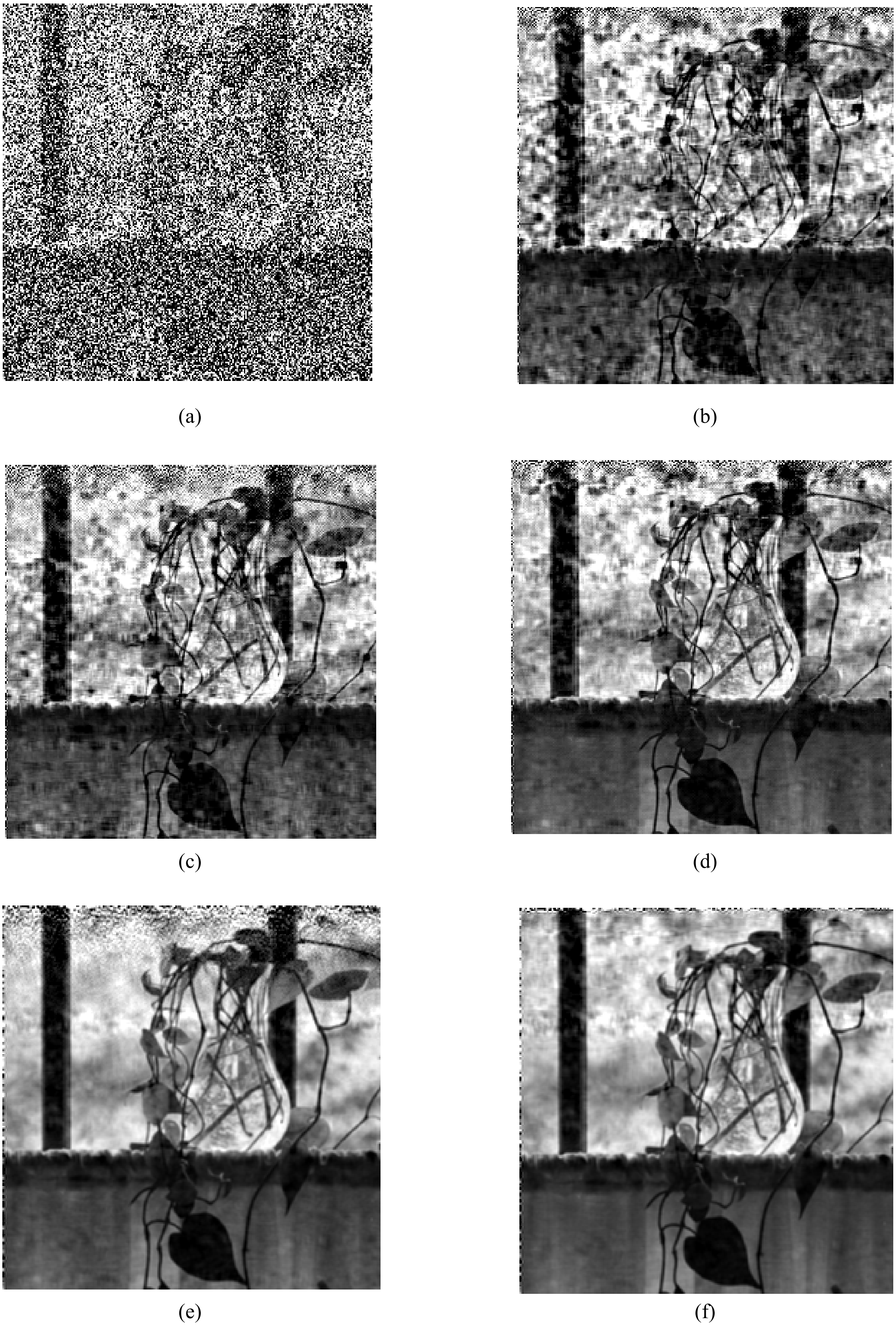
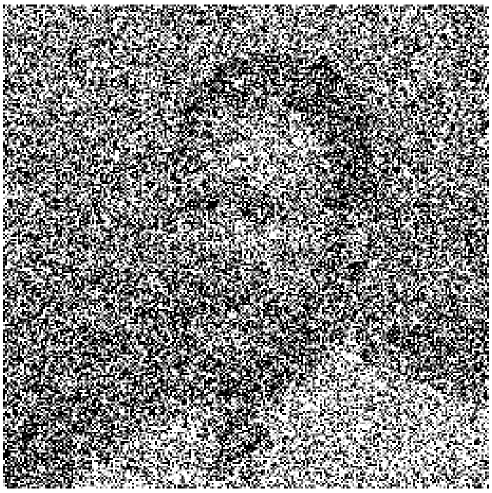


Fig. 7 Noisy and restored images by different 2D adaptive filter algorithms.
(a) Noisy image, Restored images by (b) 2D-LMS (c) 2D-NLMS (d) 2D-VSS-NLMS (e) 2D-APA (f) 2D-VSS-APA.



(a)

(b)

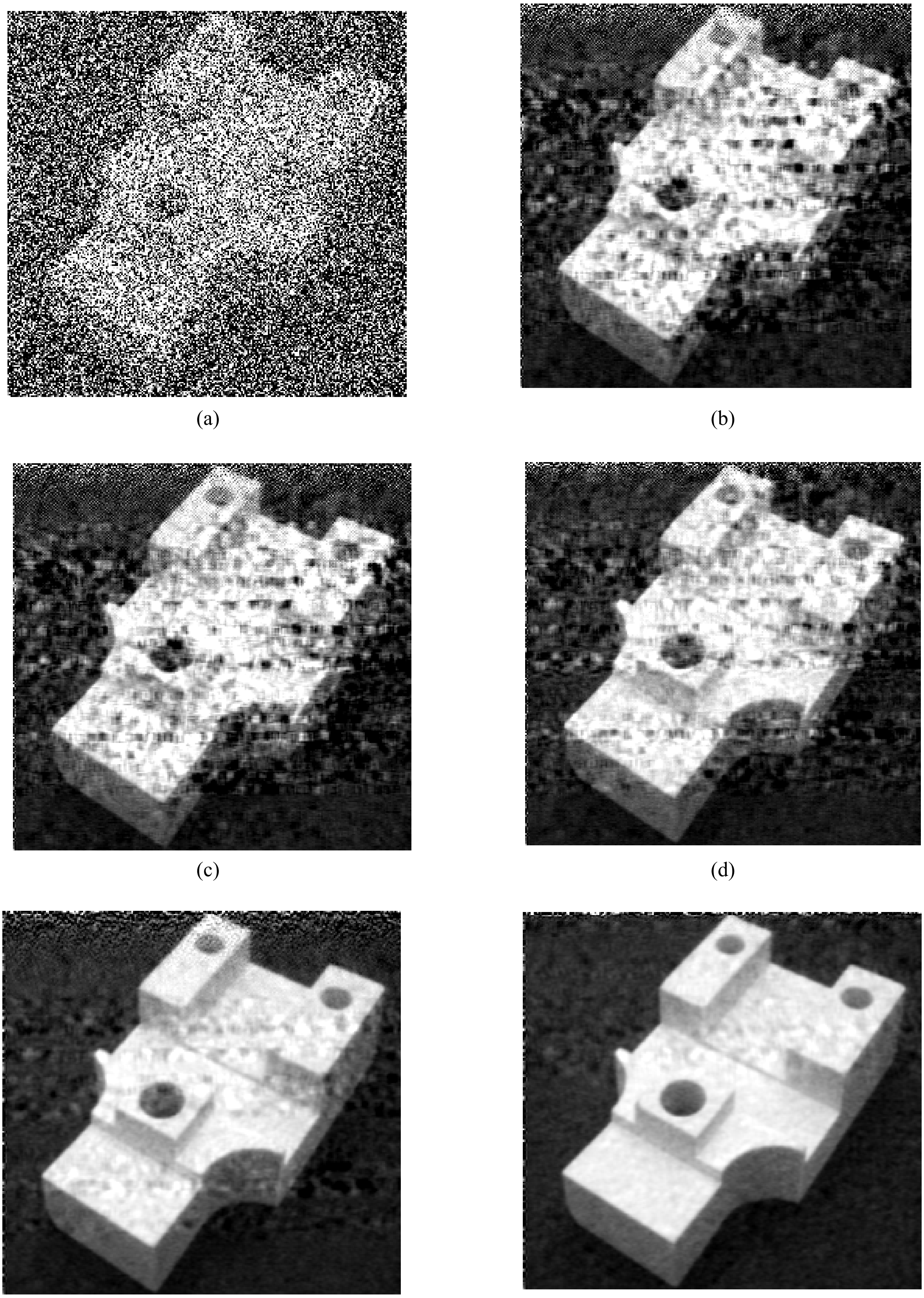
(c)

(d)

(e)

(f)

Fig. 8 Noisy and restored images by different 2D adaptive filter algorithms.
(a) Noisy image, Restored images by (b) 2D-LMS (c) 2D-NLMS (d) 2D-VSS-NLMS (e) 2D-APA (f) 2D-VSS-APA.



(a) (b)

(c) (d)

(e) (f)

Fig. 9 Noisy and restored images by different 2D adaptive filter algorithms.
(a) Noisy image, Restored images by (b) 2D-LMS (c) 2D-NLMS (d) 2D-VSS-NLMS (e) 2D-APA (f) 2D-VSS-APA.

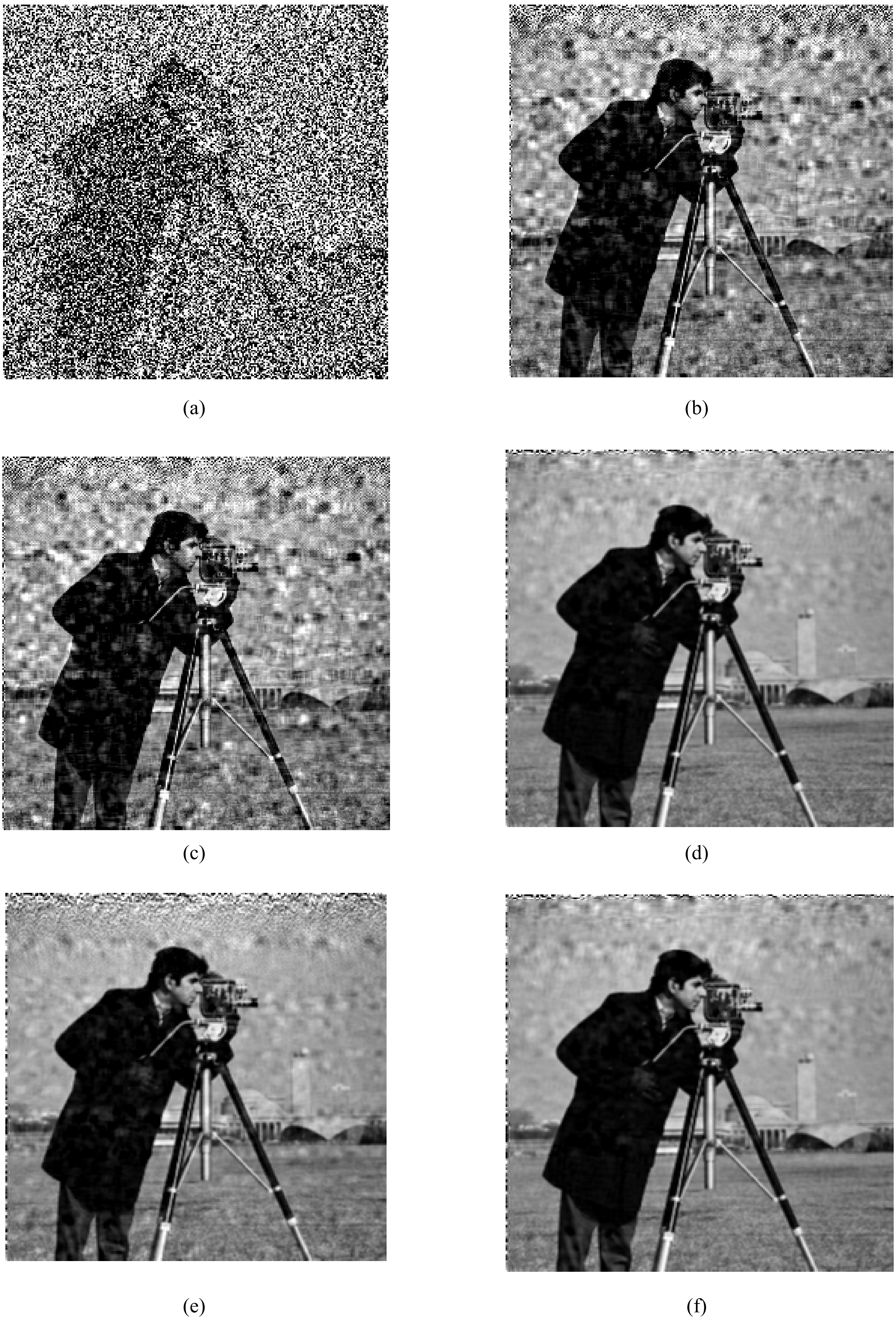
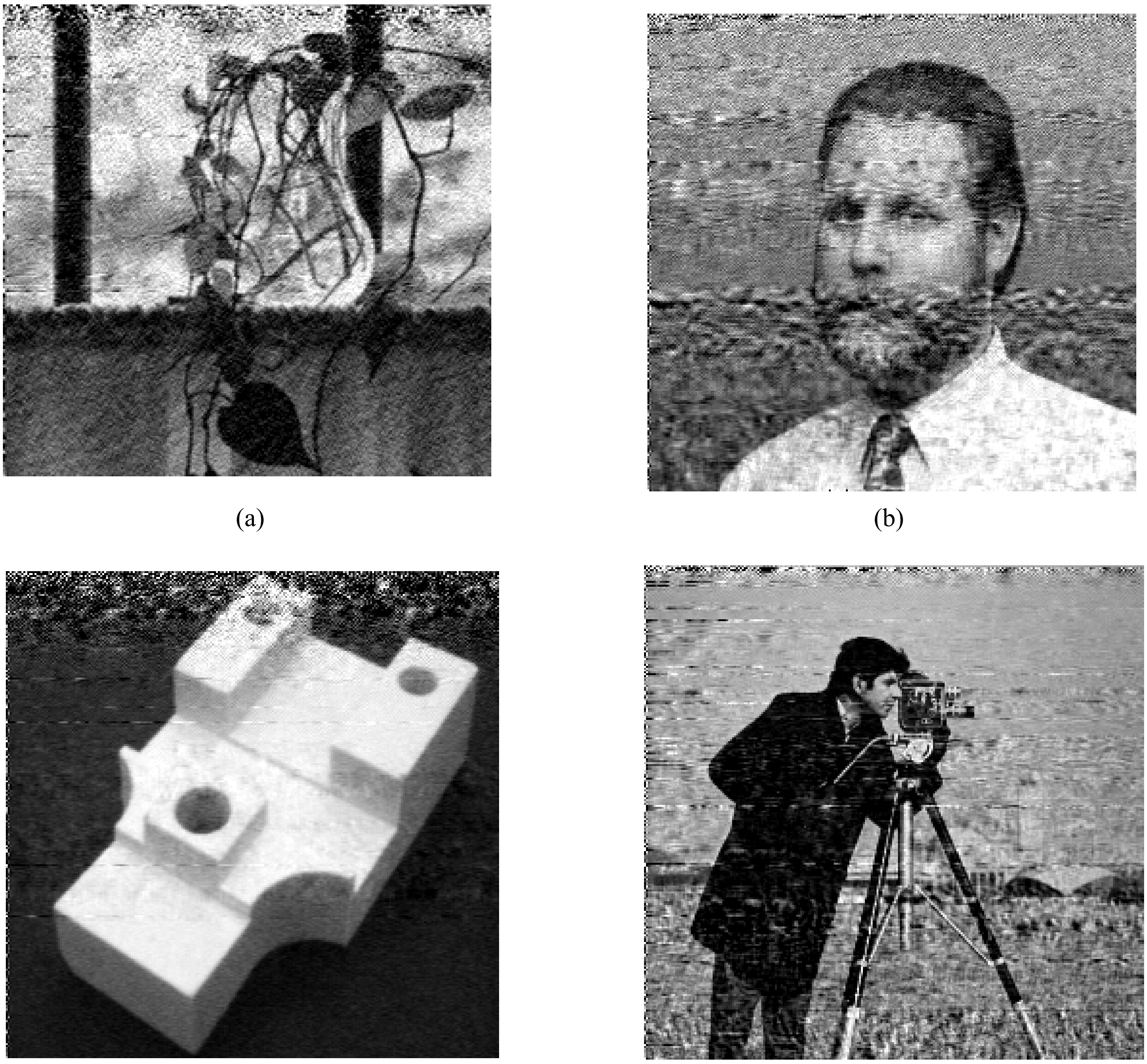


Fig. 10 Noisy and restored images by different 2D adaptive filter algorithms.
(a) Noisy image, Restored images by (b) 2D-LMS (c) 2D-NLMS (d) 2D-VSS-NLMS (e) 2D-APA (f) 2D-VSS-APA.



(a) (b)

(c) (d)

Fig. 11 Restored images for different images with 2D-SM-NLMS.

Table 5 Executing time of the proposed algorithms in 2D-ANC.

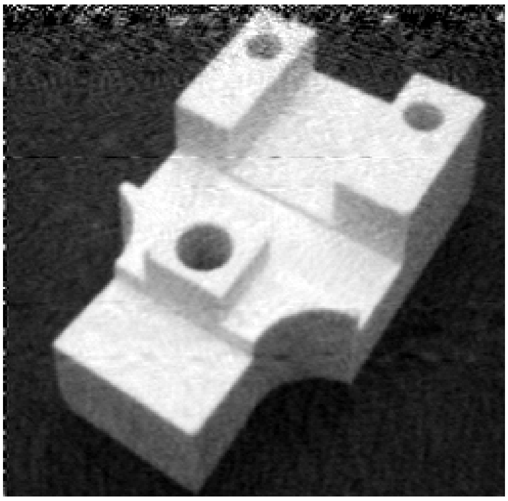
ALGORITHM	TIME(SECOND)
2D-LMS	5.87
2D-NLMS	6.31
2D-APA (K=L=2)	11.71
2D-VSS-NLMS	7.94
2D-VSS-APA(K=L=2)	13.21
2D-SPU-NLMS (S=3)	5.10
2D-SPU-NLMS (S=1)	3.31
2D-SPU-APA (K=L=2, S=3)	10.14
2D-SPU-APA(K=L=2, S=1)	8.25
2D-SM-NLMS	5.24
2D-SM-APA(K=L=2)	10.04

6 Conclusion

In this paper we presented several 2D adaptive filter algorithms. The presented algorithms are 2D-VSS-NLMS, 2D-VSS-APA, 2D-SPU-NLMS, 2D-SPU-APA, 2D-SM-NLMS, and 2D-SM-APA. The performance of these algorithms was demonstrated in 2D adaptive noise cancellation setup. The simulation results showed that the 2D-VSS adaptive filter algorithms have good ability for elimination of noise in digital images. Also, the 2D-SPU adaptive filters have low computational complexity and have close performance to classical 2D adaptive filters. In 2D-SM adaptive filters, the number of filter coefficients in update is related to the specific condition which leads to reduction in computational complexity.

(a)

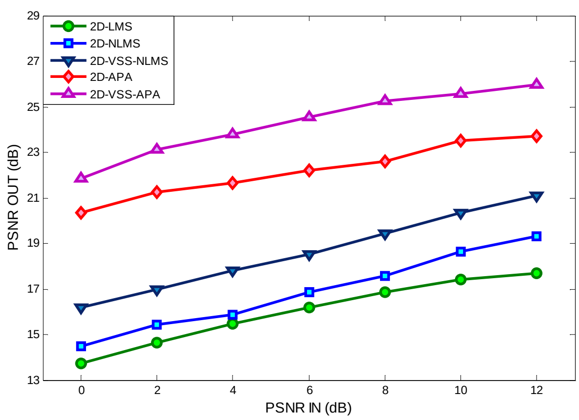
(b)



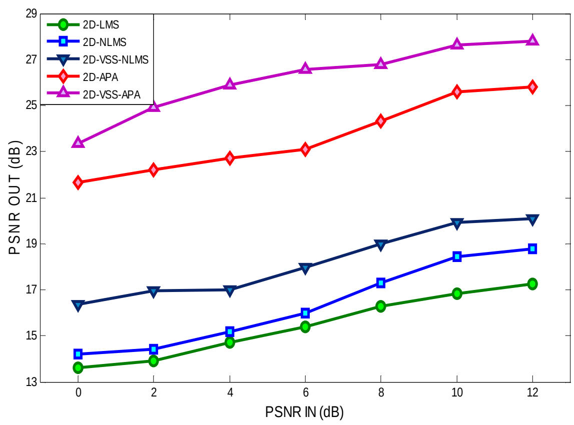
(c)

(d)

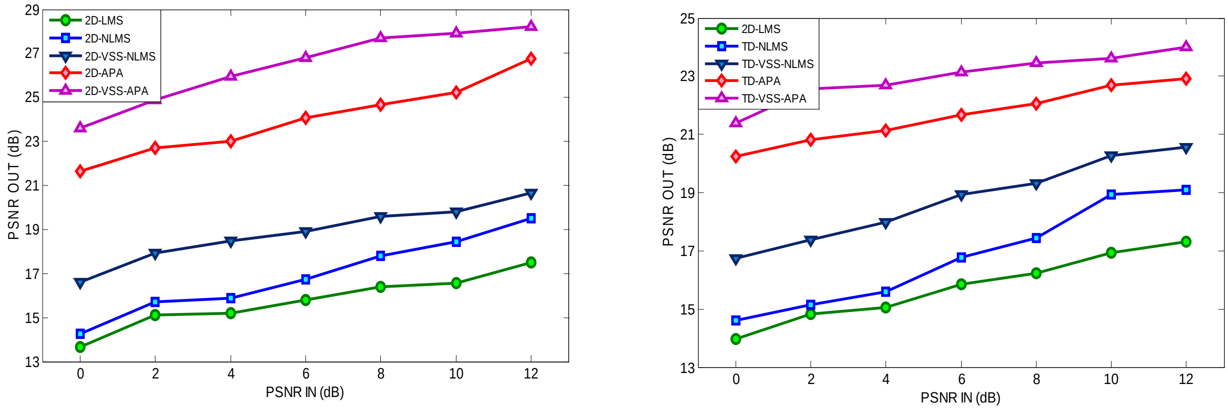
Fig. 12 Restored images for different images with 2D-SM-APA.



(a)

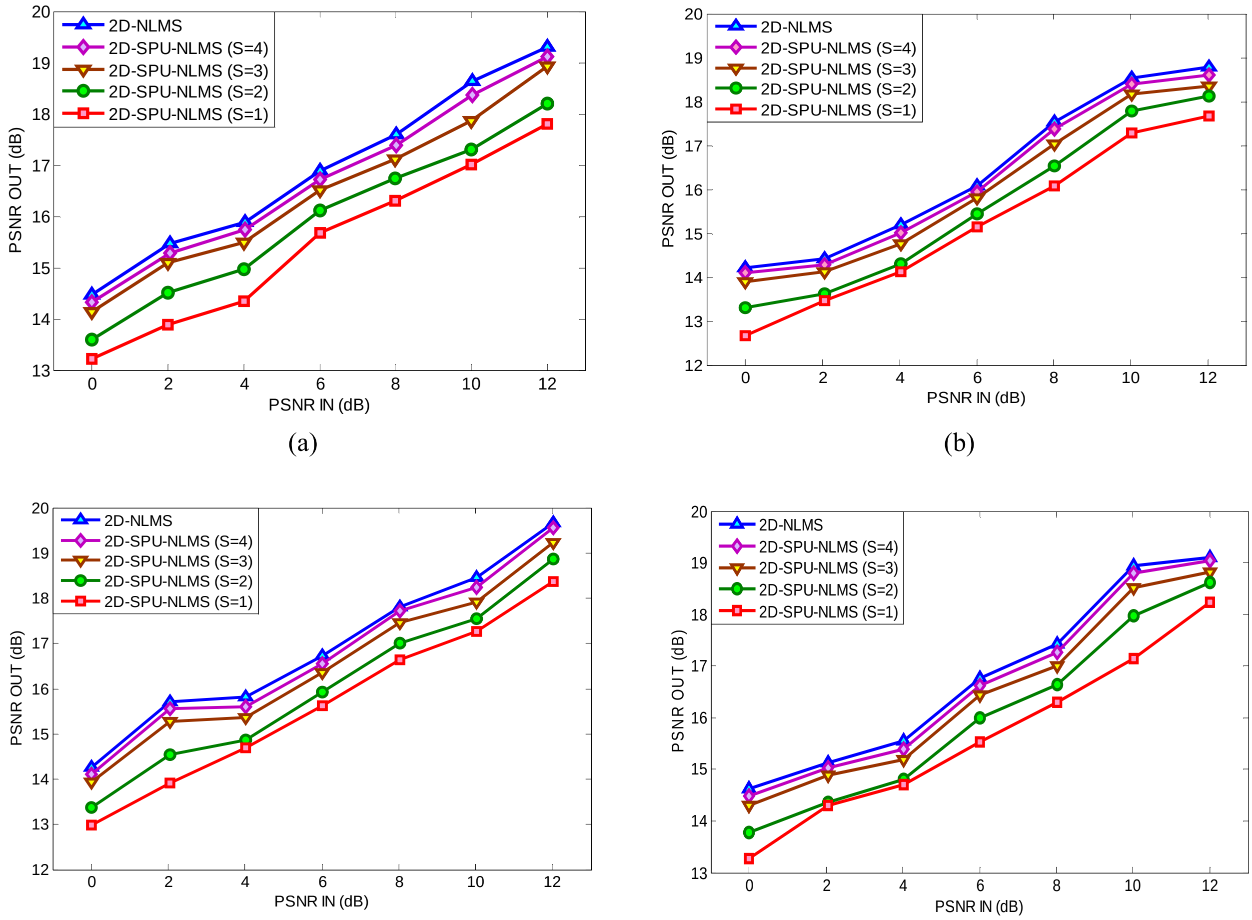


(b)



(c) (d)

Fig. 13 Output PSNR versus input PSNR for 2D-LMS, 2D-NLMS, 2D-APA, 2D-VSS-NLMS, and 2D-VSS-APA.
(a) Pot (b) Simpson (c) part (d) cameraman.



(a) (b)

(c) (d)

Fig. 14 Output PSNR versus input PSNR for 2D-SPU-NLMS with S=1, 2, 3, 4, and 5.
(a) Pot (b) Simpson (c) part (d) cameraman.

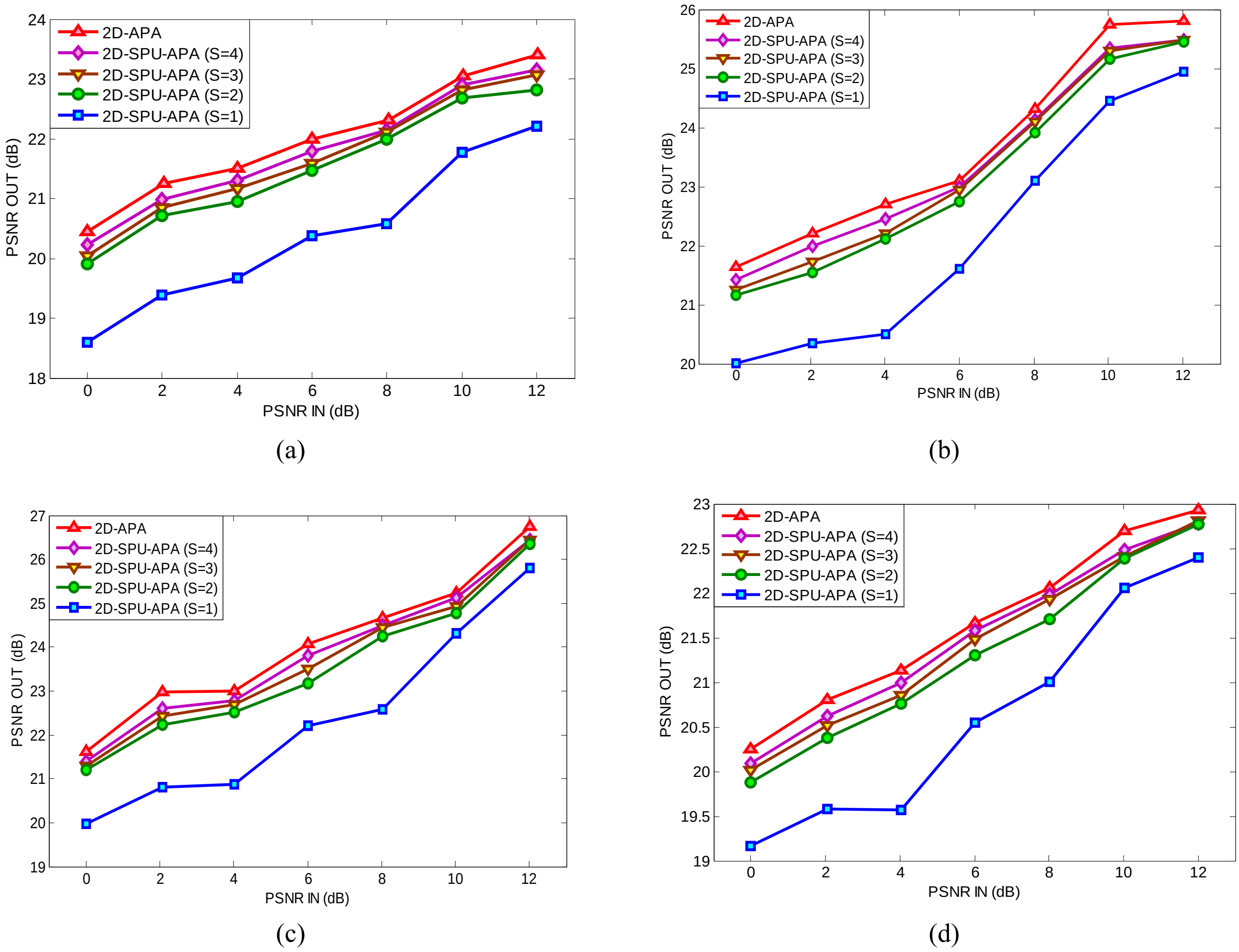


Fig. 15 Output PSNR versus input PSNR for 2D-SPU-APA with S=1, 2, 3, 4, and 5. (a) Pot (b) Simpson (c) part (d) cameraman.

Acknowledgments

The authors wish to express their thanks to the Research Institute for ICT-ITRC for their financial support for this project.

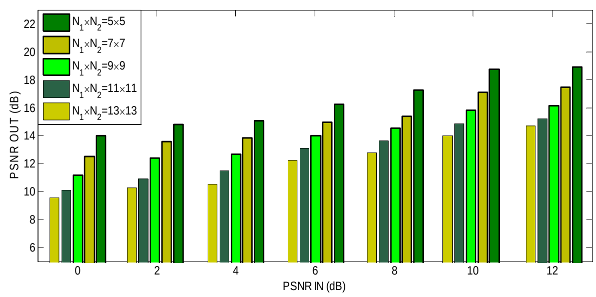


Fig. 16 Input PSNR versus output PSNR for various order of filter in 2D-LMS algorithm.

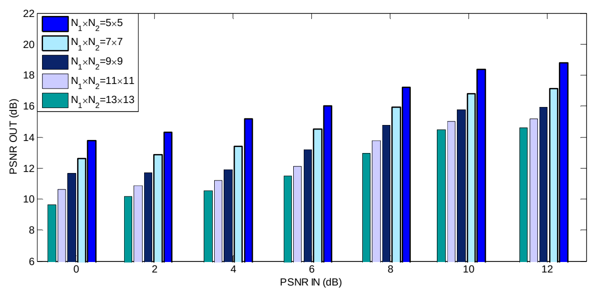


Fig. 17 Input PSNR versus output PSNR for various order of filter in 2D-NLMS algorithm.

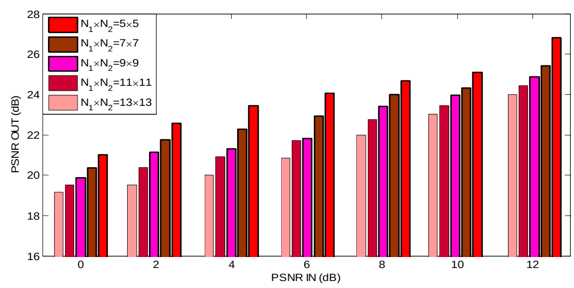


Fig. 18 Input PSNR versus output PSNR for various order of filter in 2D-APA algorithm.

References

[1] Dosaranian Moghadam M., Bakhshi H. and Dadashzadeh G., “Joint Closed-Loop Power Control and Base Station Assignment for DS-CDMA Receiver in Multipath Fading Channel with adaptive Beamforming Method”, *Iranian Journal of Electrical & Electronic Engineering* (*IJEEE*), Vol. 6, No. 3, pp. 156-167, 2010.

[2] Shams Esfand Abadi M., Mehrdad V. and Noroozi M., “A Family of Selective Partial Update Affine Projection adaptive Filtering Algorithms”, *Iranian Journal of Electrical & Electronic Engineering* (*IJEEE*), Vol. 5, No. 3, pp. 159-169, 2009.

[3] Bagheri F., Khaloozadeh H. and Abbaszadeh K., “Stator Fault Detection in Induction Machines by Parameter Estimation Using Adaptive Kalman Filter”, *Iranian Journal of Electrical & Electronic Engineering (IJEEE)*, Vol. 3, No. 3, pp. 72-82, 2007.

[4] Muneyasu M. and Hinamoto T., “A realization of 2D adaptive filters using affine projection algorithm”, *Journal of the Franklin Institute*, Vol. 335, No. 7, pp. 1185-1193, 1998.

[5] Hadhoud M. M. and Thomas D. W., “The two-dimensional adaptive LMS algorithm”, *IEEE Trans. Circuits Systems*, Vol. 35, No. 5, pp. 485-494, 1988.

[6] Jenkins W.K. and Faust R.P., “A constrained two-dimensional adaptive digital filter with reduced computational complexity”, *IEEE International Symposium on Circuits and Systems,* pp. 2655-2658, 1988.

[7] Ohki M. and Hashiguchi S., “Two-dimensional LMS adaptive filters”, *IEEE Transaction on consumer electronics*, Vol. 37, No. 1, pp. 66-73, 1991.

[8] Azimi-Sadjadi M. R. and Pan H., “Two-dimensional block diagonal LMS Adaptive filtering”, *IEEE Transactions on signal processing*, Vol. 42, No. 9, pp. 2420-2429, 1994.

[9] Glentis G.-O., “An efficient affine projection algorithm for 2D FIR adaptive filtering and linear prediction”, *Signal Processing*, Vol. 86, pp. 98-116, 2005.

[10] Muneyasu M., Uemoto E. and Hinamoto T., “A novel 2-d adaptive filter based on the 1-d RLS algorithm”, *IEEE International Symposium on Circuits and Systems*, Hong Kong, pp. 2317-2320, June 1997.

[11] Sequeria A. M. and Therrien C. W., “Anew 2-d fast RLS algorithm”, *IEEE Proc. Int. Conf. On Acoust. Speech and Signal Proc.*, Albuquerque, New Mexico, pp. 1401-1404, Apr. 1990.

[12] Liu X., Baylou P. and Najim M., “A new 2d fast lattice RLS algorithm”, *IEEE Proc. Int. Conf. On Acoust. Speech and Signal Proc.*, San Francisco, California, pp. 329-332, Mar. 1992.

[13] Mikhael W. B. and Ghosh S. M., “Two-dimensional Variable Step-Size Sequential Adaptive Gradient Algorithms with Applications”, *IEEE transaction on circuits and systems*, Vol. 38, No. 12, pp. 1577-1580, Dec. 1991.

[14] Shin H.-C., Sayed A. H. and Song W.-J., “Variable step size NLMS and affine projection Algorithm”, *IEEE signal processing letters*, Vol. 11, No. 2, pp. 132-135, 2004.

[15] Abadi M. S. E. and Husøy J. H., “Variable step-size pradhan-Reddy subband adaptive filters”, *Proc. of Fifth Intl. Conf. on Information, Communications and Signal Processing*, Bangkok, Thailand, pp. 909-912, Dec. 2005.

[16] Shams Esfand Abadi M. and Far A. M., “A unified framework for adaptive filter algorithms with variable step-size”, *Computers and Electrical Engineering*, Vol. 34, No. 3, pp. 232-249, 2008.

[17] Gollamudi S., Nagaraj S., Kapoor S. and Huang Y. F., “Set-Membership Filtering and a set-membership normalized LMS Algorithm whit an Adaptive Step-Size”, *IEEE signal processing Lett*, Vol. 5, No. 5, pp. 111-114, May 1998.

[18] Werne S. and Diniz P. S. R., “Set-Membership Affine Projection Algorithm”, *IEEE signal processing Lett,* Vol. 8, No. 8, pp. 231-235, 2001.

[19] Eweda E., “Analysis and design of a signed regressor LMS algorithm for stationary and nonstationary adaptive filtering with correlated Gaussian data,” *IEEE Trans. Circuits, Syst.*, Vol. 37, No. 11, pp. 1367-1374, Nov. 1990.

[20] Do˘ganc¸ay K. and Tanrıkulu O., “Adaptive filtering algorithms with selective partial updates”, *IEEE Trans. Circuits, Syst. II: Analog and Digital Signal Processing*, Vol. 48, No. 8, pp. 762-769, Aug. 2001.

[21] Werner S., de Campos M. L. R. and Diniz P. S. R., “Partial-update NLMS algorithms with data-selective updating”, *IEEE Trans. Signal Processing*, Vol. 52, No. 4, pp. 938-948, Apr. 2004.

[22] Schertler T., “Selective block update NLMS type algorithms”, *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Seattle, WA, pp. 1717-1720, May 1998.

[23] Werner S., de Campos M. L. R. and Diniz P. S. R., “Partial-update NLMS algorithms with data-selective updating,” *IEEE Trans. Signal Processing*, Vol. 52, No. 4, pp. 938-948, Apr. 2004.

[24] Douglas S. C., “Analysis and implementation of the max-NLMS adaptive filter”, *Proc. 29th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, pp. 659-663, Oct. 1995.

[25] Shams Esfand Abadi M. and Husøy J. H., "Mean-square performance of adaptive filters with selective partial update", *Signal Processing*, Vol. 88, No. 8, pp. 2008-2018, 2008.
[26] Shams Esfand Abadi M. and Palangi H., "Mean-square performance analysis of the family of selective partial update and selective regressor affine projection algorithms", *Signal Processing*, Vol. 90, No. 1, pp. 197-206, 2010.
[27] Widrow B. and Stearns S. D., *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1985.
[28] Haykin S., *Adaptive Filter Theory*, NJ: Prentice-Hall, 4th edition, 2002.
[29] Sayed A. H., *Fundamentals of Adaptive Filtering,* Wiley, 2003.
[30] Knuth D. E., *Sorting and Searching vol. 3 of The Art of Computer Programming*, 2nd ed. Reading, MA: Addison-Wesley, 1973.

Mohammad Shams Esfand Abadi was born in Tehran, Iran, on September 18, 1978. He received the B.S degree in Electrical Engineering from Mazandaran University, Mazandaran, Iran and the M.S. degree In Electrical Engineering from Tarbiat Modares University, Tehran, Iran in 2000 and 2002, respectively, and the Ph.D. degree in Biomedical Engineering from Tarbiat Modares University, Tehran, Iran, 2007. Since 2004 he has been with the Faculty of Electrical and Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran. During the fall of 2003, spring 2005, and again in the spring of 2007, he was a visiting scholar with the Signal Processing Group at the University of Stavanger, Norway. His research interests include digital filter theory and adaptive signal processing algorithms.

Sahar Nikbakht was born in Bandar Anzali, Iran, in 1984. She received the B.S. degree in Electrical Engineering from Lahijan University, Lahijan, Iran, in 2005. She is the master student in the Shahid Rajaee Teacher Training University, Faculty of Electrical and Computer Engineering, Tehran, Iran. Her research activities include adaptive signal processing and adaptive filtering.