Adaptive Order Statistic Filters for the Removal of Noise from Corrupted Images

S. Tsekeridou C. Kotropoulos I. Pitas

Department of Informatics

Aristotle University of Thessaloniki

Box 451, Thessaloniki 540 06, GREECE

Tel: +30-31-996304, Fax: +30-31-996304

E-mail: {sofia, costas, pitas}@zeus.csd.auth.gr

Abstract

Adaptive order statistic filters for noise smoothing in digital images are presented in this paper. Two classes of adaptive filters are studied, namely, the LMS-based adaptive order statistic filters and the signal-adaptive filters. The filter structures in the first class need a noise-free image to be used as a reference image whereas those in the second class do not require a reference image. Two filter structures from the former class are examined: the *adaptive location-invariant L-filter* and the *adaptive Ll-filter*. A novel signal-adaptive filter, namely, the *morphological signal-adaptive median filter* ($MSAM$) is proposed in the second class. It employs an anisotropic window adaptation procedure based on mathematical morphology operations. The noise smoothing capabilities and the computational complexity of the LMS-based adaptive order statistic filters studied serves as a baseline in the assessment of the properties of the proposed $MSAM$ filter. Quantitative criteria (e.g., the signal-to-noise ratio, the peak signal-to-noise ratio, the mean absolute error and the mean squared error) as well as qualitative criteria (e.g., the perceived visual quality of the processed images) are employed to assess the performance of the filters in various corruption cases by different noise models.

Keywords: noise smoothing, order statistics, LMS-based adaptive filtering, signal-adaptive filtering, morphological operations.

1 Introduction

Adaptive signal processing has been successfully applied in identification, channel equalization, echo cancellation in telephone channels, suppression of narrowband interference in wideband signals and adaptive arrays for more than two decades [1]. The most widely known adaptive filters are the linear ones that have the form of either finite impulse response (FIR) filters or lattice filters. However, linear filters may not be suitable for applications where the transmission channel is nonlinear or the noise is impulsive or the signal is strongly nonstationary (e.g. in image processing).

On the contrary, a multitude of nonlinear techniques has proved a successful alternative to the linear techniques in all the above-mentioned cases. For a review of the nonlinear filter classes the reader may consult [2]. One of the best known nonlinear filter families is based on order statistics [3]. It uses the concept of sample ordering. There is now a multitude of nonlinear filters based on data ordering. Among them are the L-filters whose output is defined as a linear combination of the order statistics of the input sequence [4]. The L-filters have found extensive application in digital signal and image processing, because they have a well-defined design methodology as the estimators which minimize the Mean-Squared Error (MSE) between the filter output and the noise-free signal. Another structure that is a generalization of L-filters are the Ll-filters that employ both time/spatial and ordering information [5]. An extension of the Ll-filters, the so-called permutation filters that embody the full potential of a permutation group transform, is proposed in [6].

It is well-known that digital image filtering techniques must take into account the local image content (i.e., the local statistics), because image statistics vary throughout an image. It has been proved both in theory and in practice that adaptive techniques can cope with nonstationary and/or time varying signals. Two broad classes of adaptive filters are known in the literature. Adaptive filters whose coefficients are determined by *iterative algorithms* (e.g. Least Mean Squares, Recursive Least Squares) for the minimization of the mean squared error (MSE) between the filter output and the desired response belong in the first class. Several authors have used the Least Mean Squares (LMS) algorithm to design nonlinear filters. For example, the LMS algorithm has been used in the design of L-filters [7, 8, 9, 10], Ll-filters [11], permutation filters [12] and order statistic filter banks [13]. A survey on adaptive order statistic filters can also be found in [14]. The *so-called signal-adaptive filters* belong to the second class. They are filters that change their smoothing properties at each image pixel according to the local image content. They have been used in image processing applications where impulsive or signal-dependent noise is present [15, 16]. Most of these filters change their coefficients close to the image edges or close to impulses, so that their performance becomes similar to that of the median filter.

In this paper, a comparative study of the noise suppression capability of filter structures that belong to the above-mentioned classes is undertaken. Still images are examined. However, a straightforward extention to the case of image sequences is possible as well. In the first class, adaptive order statistic filters based on LMS are considered. Two filter structures are discussed: (1) the location-invariant adaptive LMS L-filter [8, 10] and (2) the adaptive LMS Ll-filter [11]. Both adaptive L- and Ll-filters depend on the availability of a noise-free image to be used as a reference image. It is known that the location-invariant LMS L-filter, in principle, can be modified so that it minimizes the overall output filter, thus alleviating the need for having a reference image [10]. When a reference image is not available, another possibility is to use a filter that belongs to the second class, i.e., a signal-adaptive filter. The smoothing properties of signal-adaptive filters change at each image pixel according to a local SNR measure that is used to vary the filter window size as well. In this paper, we shall confine ourselves to the signal-adaptive median (SAM) filter. A novel extension of the SAM filter is proposed. It is the so-called *morphological signal-adaptive median* ($MSAM$) filter. Two modifications are introduced in the SAM filter aiming at alleviating its disadvantages/limitations: (1) the impulse detection mechanism of SAM is enhanced so that it detects not only impulses of a constant amplitude but randomly-valued impulses as well, (2) in contrast to SAM that employs isotropic filter windows of dimensions 3×3 to 11×11, $MSAM$ implements anisotropic window adaptation based on binary morphological erosions/dilations with predefined structuring elements. The $MSAM$ filter performs well against contaminated Gaussian or impulsive noise corruption cases. It does not require the a priori knowledge of a noise-free image, but only of certain noise characteristics which can easily be estimated. It adapts its behavior based on a local SNR measure thus achieving edge preservation and noise smoothing in homogeneous regions.

The following commonly used objective criteria, i.e., the Signal-to-Noise Ratio (SNR), the Peak Signal-to-Noise Ratio ($PSNR$), the Mean Squared Error (MSE) and the Mean Absolute Error (MAE) have been employed in the comparisons undertaken. In addition to the quantitative criteria, subjective criteria (e.g., the visual quality perceived by the observer) in judging the success of a filtering algorithm are also used. The importance of subjective criteria becomes evident in the case of the morphological SAM filter whose output is of better visual quality (as is shown in subsequent sections) than the output of the other algorithms, although the $SNR/PSNR/MAE/MSE$ values achieved are not always the best.

In summary, the main contribution of the paper is in (a) the comparison of the adaptive nonlinear filters under study for noise suppression in still images, and (b) the design of morphological signal-adaptive median filters.

The outline of the paper is as follows. Section 2 is devoted to a brief description of the LMS adaptive order statistic filters for noise suppression in still images. The morphological signal-adaptive median filter is described in Section 3. Experimental results are included in Section 4 and conclusions are drawn in Section 5.

2 Still image adaptive LMS order statistic filtering

Let the observed image $x(\mathbf{k})$ be expressed as the sum of a noise-free image $d(\mathbf{k})$ plus zero-mean two-dimensional additive white noise, i.e., $x(\mathbf{k}) = d(\mathbf{k}) + \eta(\mathbf{k})$ where $\mathbf{k} = (k, l)$ denotes the pixel coordinates. In image processing, a neighborhood is defined around each pixel $\mathbf{k}$. Our purpose is to design a filter defined on this neighborhood (to be called the filter window hereafter) that aims at estimating the noise-free central image pixel value $d(\mathbf{k})$ by minimizing a certain criterion. Among the several filter masks (e.g. cross, x-shape, square, circle) that are used in digital image processing [2], we shall rely on the square window of dimensions $(2\xi+1) \times (2\xi+1)$. Let $N = (2\xi+1)^2$. Since we intend to apply a filter based on sample ordering, let us rearrange the above $(2\xi+1) \times (2\xi+1)$ filter window in a lexicographic order (i.e., row by row) to a $N \times 1$ vector:

$$\mathbf{x}(\mathbf{k}) = \left(x(k-\xi, l-\xi), x(k-\xi, l-\xi+1), \ldots, x(k-\xi, l+\xi), \ldots, x(k+\xi, l+\xi)\right)^T. \quad (1)$$

where T denotes the transposition operator. If $\mathcal{K}$ and $\mathcal{L}$ denote the image rows and columns respectively, depending on the image scanning method, each pixel (k, l), $k = 1, \ldots, \mathcal{K}$, $l = 1, \ldots, \mathcal{L}$ can be represented by a single running index n. For raster scan (or lexicographic order), the indices k, l and the running index n are related through $n = (k-1)\,\mathcal{L} + l$. Henceforth, a one-dimensional (1-D) notation is adopted for simplicity. The location-invariant LMS L-filter and the adaptive LMS Ll-filter are briefly reviewed in Section 2-A and Section 2-B, respectively.

2-A Location-invariant LMS L-filter

2-A.1 Filter Design

Let $\mathbf{x}_r(n)$ be the ordered input vector at pixel n given by:

$$\mathbf{x}_r(n) = \Big(x_{(1)}(n), x_{(2)}(n), \ldots, x_{(N)}(n)\Big)^T \quad (2)$$

where $x_{(1)}(n) \leq x_{(2)}(n) \leq \ldots \leq x_{(N)}(n)$. We are seeking the L-filter whose output at n, $y(n) = \mathbf{a}^T\,\mathbf{x}_r(n)$ minimizes the MSE $J(n) = \mathrm{E}\left[(y(n) - d(n))^2\right]$ subject to the constraint $\mathbf{1}_N^T\,\mathbf{a}(n) = 1$.

Let $\nu = (N+1)/2$. By employing the above-described constraint, we can partition the L-filter coefficient vector as follows:

$$\mathbf{a}(n) = \left(\mathbf{a}_1^T(n) | a_\nu(n) | \mathbf{a}_2^T(n)\right)^T \tag{3}$$

where $\mathbf{a}_1(n)$, $\mathbf{a}_2(n)$ are $(N-1)/2 \times 1$ vectors given by:

$$\mathbf{a}_1(n) = (a_1(n), \ldots, a_{\nu-1}(n))^T \qquad \mathbf{a}_2(n) = (a_{\nu+1}(n), \ldots, a_N(n))^T \,. \tag{4}$$

The coefficient for the median input sample is evaluated then by:

$$a_\nu(n) = 1 - \mathbf{1}_{\nu-1}^T \mathbf{a}_1(n) - \mathbf{1}_{\nu-1}^T \mathbf{a}_2(n). \tag{5}$$

Let $\mathbf{a}'(n)$ be the reduced L-filter coefficient vector:

$$\mathbf{a}'(n) = \left(\mathbf{a}_1^T(n) | \mathbf{a}_2^T(n)\right)^T \tag{6}$$

and $\hat{\mathbf{x}}_r(n)$ be the $(N-1) \times 1$ vector:

$$\hat{\mathbf{x}}_r(n) = \begin{bmatrix} \mathbf{x}_{r1}(n) - x_{(\nu)}(n)\mathbf{1} \\ \mathbf{x}_{r2}(n) - x_{(\nu)}(n)\mathbf{1} \end{bmatrix} . \tag{7}$$

Following the analysis in [8, 10], it can be proved that the LMS recursive relation for updating the reduced L-filter coefficient vector is given by

$$\hat{\mathbf{a}}'(n+1) = \hat{\mathbf{a}}'(n) + \mu\, \varepsilon(n)\, \hat{\mathbf{x}}_r(n) \tag{8}$$

where $\varepsilon(n)$ is the estimation error at pixel n, i.e., $\varepsilon(n) = y(n) - d(n)$ and μ is the step-size parameter considered equal to 5×10^{-7}. Eq. (5) and (8) define the LMS location invariant L-filter.

2-A.2 Complexity-Storage Requirements

The implementation of a location-invariant LMS L-filter requires a total number of $2N$ multiplications/divisions (or mults/divs) and $4(N-1)$ additions/subtractions (or adds/divs) per output sample. Thus, the filter exhibits a complexity of $O(N)$. Memory requirements are limited to the use of $3N+1$ local variables per output sample. An algorithm for fast sorting has been proposed in [17], which may be used for the input vector ordering. This method requires only $2(N-1)-log_2 N$ comparisons while the QUICKSORT method exhibits a complexity of $O(2Nlog(2\xi+1))$. L-filters are known for their small computational complexity and limited memory storage requirements.

2-B Adaptive LMS Ll-filters

2-B.1 Filter Design

The main problem with L-filters is that they do not incorporate time or spatial information. On the contrary, Ll-filters can incorporate time/spatial information along with ordering information through the representation of the observations by means of the $(N^2 \times 1)$ vector $\mathbf{z}$ [5]:

$$\mathbf{z}(n) = \left(\underline{\zeta}_1^T(\mathbf{x}(n)), \underline{\zeta}_2^T(\mathbf{x}(n)), \ldots, \underline{\zeta}_N^T(\mathbf{x}(n))\right)^T \tag{9}$$

where each $(N \times 1)$ vector $\underline{\zeta}_i^T(\mathbf{x})$ is defined by:

$$\underline{\zeta}_i^T(\mathbf{x}) = \left(x_{(i)1}, x_{(i)2}, \ldots, x_{(i)N}\right)^T, \quad i = 1, \ldots, N \tag{10}$$

with

$$x_{(i)j} = \begin{cases} x_{(i)}, & \text{if } x_{(i)} = x(j), \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

By its construction, vector $\mathbf{z}(n)$ contains both time and ordering information. It is worth noting that only N elements of vector $\mathbf{z}(k)$ are always nonzero. The output of the Ll-filter is given by: $y(n) = \mathbf{c}^T\mathbf{z}(n)$ where $\mathbf{c}$ is a $(N^2 \times 1)$ coefficient vector of the form:

$$\mathbf{c} = \left(c_{(1)1}, \ldots, c_{(1)N} | c_{(2)1}, \ldots, c_{(2)N} | \ldots | c_{(N)1}, \ldots, c_{(N)N}\right)^T. \tag{12}$$

The LMS algorithm can be used for the adaptation of the Ll-filter coefficients yielding the following updating equation [11]:

$$\mathbf{c}(n+1) = \mathbf{c}(n) + \mu\varepsilon(n)\mathbf{z}(n) \tag{13}$$

where the adaptation step-size μ should satisfy:

$$0 \leq \mu \leq \frac{2}{N\sigma^2_{\mathbf{x}_n}} \tag{14}$$

with $\sigma^2_{\mathbf{x}_n}$ denoting the total input power.

2-B.2 Complexity-Storage Requirements

For the calculation of an output sample, a total number of $2N^2+1$ mults/divs and $2N^2-1$ adds/subs are required, resulting to a complexity of $O(N^2)$. Storage requirements are quite extensive reaching a total number of $2N^2+2$ local variables per output sample. The number of comparisons needed for the input vector ordering are identical to those required by a location invariant L-filter, if the same filter window is employed. Fast sorting algorithms could also be used in this case for ordering the input.

3 Morphological signal adaptive median filter

3-A Filter Design

The filter structures described so far require the a priori knowledge of the noise-free version of the corrupted image (to be used in the training phase, as it will be described later on). However, the assumption of the availability of the noise-free version of the corrupted image is not always true. A class of filters that does not require the a priori knowledge of the desired image is the so-called *signal-adaptive filters*. Signal-adaptive filters adapt their behavior in accordance with a local SNR measure. Thus, they perform differently close to image edges, where the primary goal is to preserve them, than in homogeneous regions, where the objective is to smooth the noise. The signal adaptive median filter (SAM) is a paradigm of this class [18]. Other signal adaptive filters have been proposed in [16, 19].

A novel extension of the standard SAM filter is introduced in this section. It is the so-called *morphological signal-adaptive median filter* ($MSAM$). The proposed filter performs well on when impulsive or additive contaminated Gaussian noise is present, especially at high corruption cases. It requires only the a priori knowledge of certain noise characteristics, which can easily be estimated. It adapts its behavior based on a local SNR measure thus achieving edge preservation and noise smoothing in homogeneous regions.

To begin with, let us describe the framework for signal-adaptive filters. Let us denote by σ_η^2 the noise variance that is known or has been estimated beforehand. Moreover, in the case of impulsive noise let p_p be the percentage of positive impulses (i.e., $S_{max} = 255$), and p_η be the percentage of negative impulses (i.e., $S_{min} = 0$). Thus, the noisy image pixel values $x(n)$ are determined by the model:

$$x(n) = \begin{cases} S_{min} & \text{with probability } p_\eta \\ S_{max} & \text{with probability } p_p \\ x(n) & \text{with probability } 1 - (p_p + p_\eta) \end{cases} \tag{15}$$

where $x(n)$ is an image pixel corrupted possibly by additive white or signal-dependent noise. The output of the $MSAM$ filter is expressed as in the standard SAM filter [18], i.e.,

$$y(n) = \tilde{x}_M(n) + b(n)[x(n) - \tilde{x}_M(n)]. \tag{16}$$

$\tilde{x}_M(n)$ is the modified median, i.e., the median of the pixels that remain after the removal of impulses from the local window. $b(n)$ is a weighting coefficient that is used to adapt the window size according to whether a flat region or an edge has been met. Thus, in homogeneous regions, total noise suppression is achieved, because a large window is employed due to $b(n)$ being close to 0. Edges are also well preserved, because a small window size is used due to $b(n)$ being close to 1 in this case. The window increment/decrement procedure is explained below. Two major modifications in the standard SAM filter [18] are introduced by the so-called $MSAM$ filter:

1. $MSAM$ employs the morphological operations of dilation and erosion with certain predefined structuring elements (SEs), in order to vary anisotropically the window size with respect to the local image content. This modification has been motivated by the work reported in [20].

2. *MSAM* employs two impulse detectors: one for constant impulses (either positive or negative) and another for randomly-valued impulses. Impulse detection is done only in the initial window of dimensions 3 × 3.

Subsequently, the several steps of the algorithm are presented.

1. *Constant value impulse detection* [18]:

The filter performs detection of constant value impulses in an initial window of dimensions 3 × 3 by using a signal-dependent threshold $\tau_\eta(n)$ for negative impulses given by:

$$\tau_\eta(n) = c[S_{min} - \tilde{x}_M(n)] \leq 0 \tag{17}$$

and another one for positive impulses defined by:

$$\tau_p(n) = c[S_{max} - \tilde{x}_M(n)] \geq 0 \tag{18}$$

where $c = 5/6$. If $[x(n) - \tilde{x}_M(n)] < \tau_\eta$ then $x(n)$ is detected as a negative impulse. On the opposite, if $[x(n) - \tilde{x}_M(n)] > \tau_p$ then $x(n)$ is detected as a positive impulse.

2. *Randomly-valued impulse detection:*

Motivated by the randomly-valued impulse detection mechanisms developed in [16, 21], two additional thresholds are incorporated in the standard *SAM* filter. They are defined as follows:

$$h_1(n) = x_{max} - x_{min_2} \tag{19}$$

$$h_2(n) = x_{max_2} - x_{min} \tag{20}$$

where x_{min} is the minimum value pixel, x_{min_2} the second minimum value pixel, x_{max} the maximum value pixel and x_{max_2} the second maximum value pixel in the initial window. If the absolute difference $|x(n) - \tilde{x}_M(n)|$ is greater than any of the thresholds h_1 or h_2, then $x(n)$ is of very small or of very large value with respect to its neighboring pixels and most possibly is a randomly-valued impulse. Random-value impulse detection is performed only if the current pixel has not previously been detected as a constant impulse and only when $b_{3\times3}(n) > b_t$ (which means edge region or possible randomly-valued impulse in the initial 3×3 window with respect to Steps 3 and 4 described later on). If the current pixel is an impulse, either constant or randomly-valued, it is excluded from the estimation of the median at the current and at any future window centered at n. That is, it is not considered in the estimation of the modified median employed in (16).

3. *Calculation of the weighting coefficient* $b(n)$*.*

This coefficient is given by the expression [18]:

$$b(n) = \begin{cases} 0, & \text{if } \alpha\sigma_\eta^2 \geq \hat{\sigma}_x^2 \\ (1 - \alpha\frac{\sigma_\eta^2}{\hat{\sigma}_x^2})^\beta, & \text{otherwise.} \end{cases} \tag{21}$$

$\hat{\sigma}_x^2$ denotes the image variance estimated from the local "windowed" histogram [18] from which the current pixel is additionally excluded if it is detected as an impulse. α and β are appropriately chosen parameters in the interval $[0,\ 1]$. The parameter α controls the threshold on the local signal to noise ratio up to which the high-frequency components are entirely suppressed. The parameter β controls the suppression of noise close to edges.

4. *Decision whether the current pixel belongs to a homogeneous region or to an edge.*

The weighting factor $b(n)$ calculated in Step 3 is compared to a predefined threshold b_t. If it is smaller than b_t, then the current pixel is assumed to belong to a homogeneous region. Otherwise, the current pixel belongs to an edge. The threshold b_t lies in the interval $[0,\ 1]$. Its selection is accomplished in accordance with the degree of corruption and the nature of noise. For highly corrupted images, its value is lower than 0.5. If the image is corrupted by pure Gaussian noise of relatively medium variance, the threshold lies in the range $[0.65,\ 0.85]$. A reliable method for the choice of the threshold b_t is described in [18]. Methods employing local statistics have been reported in [22].

5. *Novel window adaptation procedure.*

The proposed $MSAM$ further differs from the SAM filter in the window adaptation procedure used. SAM employs isotropic filter windows of dimensions 3×3 up to 11×11. In contrast to SAM, an anisotropic window adaptation procedure is proposed based on mathematical morphology operations of binary erosion/dilation with predefined structuring elements. Four structuring elements are employed, namely, B_1, B_2, B_3 and B_4 and their symmetric ones B_1^s, B_2^s, B_3^s and B_4^s illustrated in Figure 1a. They are divided in even-angle SEs (B_1, B_2, B_1^s, B_2^s) and in odd-angle SEs (B_3, B_4, B_3^s, B_4^s). The window increment is performed by a dilation operation $W \oplus B_i$, where W denotes the current filter window. The "direction" of increment depends on the choice of B_i. The result of the window growing $W \oplus B_i$ for an original 3×3 window size W is shown in Figure 1a. The thin dots belong to the original window W while the bold dots denote the new pixels that have been appended to W to form a new (larger) window. In a similar fashion, the window decrement is performed by an erosion operation $W \ominus B_i$. This is demonstrated in Figure 1b for B_1, B_1^s, B_2, B_2^s. It should be noted that only these SEs are used for reducing the window size. The procedure of the window adaptation begins with a 3×3 square window and checks whether the central pixel belongs to an edge.

I. If it does not belong to an edge:

(a) An attempt is made to increase the window size by using the odd-angle SEs.

(i) If an edge is **"hit"** (e.g. $b(n) > b_t$) the current odd-angle SE is excluded and the even-angle SEs that compose the SE under consideration are tested for possible window increment. For example, if B_4 is excluded, B_1 and B_2 are tested for possible use in window increment. This means that the sides of the mask are also separately checked expecting that an edge is possibly met at one side only, thus allowing window increment from the remaining side. By doing so, maximal window increment is achieved.

(ii) If an edge is **not** met, then the current odd-angle SE is used to increase the window size. The corresponding even-angle SEs are then excluded.

(b) In the next step, the odd-angle SEs, that have not been excluded in a previous step, are tested again. In the above-described example, B_3, B_3^s and B_4^s remain to be tested. In other words, if it is known from a previous step that a window side meets an edge, this side is not considered again.

(c) The procedure continues until all the odd-angle and all the even-angle SEs are excluded or until at least one side reaches a maximal size (e.g., 11).

II. **If the pixel belongs to an edge**, the goal is to expand the mask in the neighboring regions that are homogeneous. That is, the current pixel is labelled as a border pixel and the window increment is done towards the side of the edge where the pixel belongs to. To do so, the opposite side of the edge must be found and the increment of the filter window towards that direction must be prohibited. This is done as follows. The average value of the pixels on each of the four sides of a window of dimensions 3×3 is derived and the absolute difference between the side average values and the current pixel is calculated. The side that corresponds to the greater difference is removed. The difference is a measure of deviation of the side pixels from the current one. The side that deviates the most is possibly the side that should be removed. The decrement of the initial window size is achieved by the operation of erosion with one of the SEs B_1, B_1^s, B_2, B_2^s. Subsequently, the window increases towards the remaining sides in the way described above by using appropriate SEs. For example, if an erosion with B_1 were performed, B_1, B_2, B_2^s, B_3, B_4^s would be used to increase further the window.

Finally, if the current pixel is detected as an impulse, the factor $b(n)$ is set to 0 (thus allowing maximum filtering). After the "optimal" window size with respect to the local image content has been determined, the filter output is estimated by (16). The modified median $\tilde{x}_M(n)$ in 16 is calculated only twice: at the initial window for impulse detection and at the final window for the filter output estimation unlike the classical SAM filter which, in case that impulsive noise is present, requires median estimation at every step of the isotropic window adaptation procedure.

3-B Complexity-Storage Requirements

In order to analyse the complexity and storage needs of the $MSAM$ filter, the best and worst cases with respect to the "optimal" filter window size will be examined. It will be assumed that impulse detection is also performed, that is, impulsive noise is observed. The total number of final mults/divs and adds/subs required per output sample is dependent on the local image content, which controls the anisotropic window adaptation.

3-B.1 Best case - Minimum Complexity

The minimum number of computations is needed when the anisotropic window process outputs as "optimal" local window the initial window consisting of $N_I = n_I \times n_I$ pixels. In this case, the "windowed histogram" estimation needs 4 mults/divs and $N_I + 6$ adds/divs [18]. Impulse detection requires 2 mults/divs and 6 adds/subs. The calculation of $b(n)$ involves, assuming that $\beta = 1.0$, $N_I + 5$ mults/adds and $2N_I + 4$ adds/subs. Finally, the calculation of the output needs 2 adds/subs and 1 mult/div. The total number of mults/divs and adds/subs sums up to $N_I + 12$ and $3N_I + 18$ respectively. The additional computational cost, in this case, compared to the SAM filter, is the need of 3 adds/subs for random impulse detection (the sorting that is required is performed during the modified median estimation). For median estimation, fast algorithms have been proposed in the literature [23, 24]. If the modified median is calculated using the method of [23], only $2\lceil\sqrt{N_I - \Delta h_{min} - \Delta h_{max} - 1}\rceil$ comparisons are necessary. Δh_{min} and Δh_{max} are defined in [18].

Storage needs sum up to an array of 256 elements representing the local "windowed" histogram (assuming that $S_{max} = 255$) and another 16 local variables used for impulse detection, $b(n)$ and output estimation. The extra memory needs of the $MSAM$ filter compared with SAM are 2 local variables used by the randomly-valued impulse detector.

3-B.2 Worst case - Maximum Complexity

The worst case, in terms of computational complexity, occurs when $MSAM$ outputs as “optimal” the maximal filter window allowed consisting of $N_F = n_F \times n_F$ pixels. Let’s assume that the number of stages needed in order that the initial window $n_I \times n_I$ reaches the maximal one $n_F \times n_F$ are equal to S_T. S_D from S_T result from dilations (window increment) while the remaining S_E from erosions (window decrement), i.e. $S_T = S_D + S_E$. Furthermore, let $n_{S_D,k}$ be the number of pixels added to the local window at increment stage k, $k = 1, \cdots, S_D$, and $n_{S_E,l}$ be the number of pixels removed from the local window at decrement stage l, $l = 1, \cdots, S_E$. It is evident that the total number of necessary stages for maximal filter window for the SAM filter, $S_{T,SAM}$ is less than S_T, due to isotropic window adaptation.

The number of mults/divs and adds/subs required by the four main steps of the $MSAM$ algorithm: “windowed histogram” evaluation (includes the window adaptation process), impulse detection, $b(n)$ calculation and output estimation are summarized in Table 1 along with the number of comparisons required for modified median calculation per output sample. For comparison purposes, respective numbers are given for the SAM filter. Decrement stages involve adding and later removing pixels from the window leading to double computations ($2\sum_{j=1}^{S_E} n_{S_E,j}$) compared to increment stages. For the calculation of $b(n)$ the worst case is considered (computation of $1 - \frac{\alpha\sigma_n^2}{\sigma_x^2}$). SAM impulse detection complexity is evaluated along with $b(n)$ calculation complexity, since impulse detection is performed at every stage and directly affects the value of $b(n)$ [18]. This imposes an extra cost of $2S_{T,SAM}$ mults/divs and $3S_{T,SAM}$ adds/subs. On the contrary, $MSAM$ impulse detection is performed only at the initial window. Modified median estimation is performed by $MSAM$ only at the initial and final windows ($\Delta h_{min,I}$, $\Delta h_{max,I}$ in Table 1 correspond to the initial window while $\Delta h_{min,F}$, $\Delta h_{max,F}$ to the final one). On the other hand, SAM evaluates median values at every stage for impulse detection (in Table 1 $N_1 = N_I$ and $N_{S_{T,SAM}} = N_F$). It is noted that generally $N_I + \sum_{i=1}^{S_D} n_{S_D,i} + \sum_{j=1}^{S_E} n_{S_E,j} \geq N_F$ The equality is valid only if $S_E = 0$.

Storage requirements, in the worst case, are equal to those of the best one plus an additional number of 8 local variables indicating whether a SE has been employed for erosion or dilation in the current stage or if it has been excluded from use.

4 Simulation Results

4-A Performance Evaluation

The noise-free images "Airfield" and "Bridge" have been corrupted by adding white i.i.d. noise obeying the pdf of a Gaussian mixture given by:

$$\eta \sim (1-\lambda)N(0,\sigma_\nu) + \lambda N(0,\frac{\sigma_\nu}{\lambda}) \tag{22}$$

with mean value $E(\eta)$ close to 0, and variance $\sigma_\eta^2 = \sigma_\nu^2(1-\lambda+1/\lambda)$. The contamination factor λ, along with the initial standard deviation σ_ν, determine the degree of corruption. The result of this noise corruption process is a mixture of Gaussian and impulsive noise (contaminated Gaussian noise) of varying characteristics according to the value of both λ and σ_ν. Thus, the proposed filter performance can be evaluated on a wide range of different combinations of noise distributions and corruption levels. For more details, the reader is referred to [25]. It is worth noting that the special case of pure constant value impulsive noise is additionally examined in order to test the robustness of the filters to this kind of noise. Performance results are reported for corruption cases by additive noise with Laplacian (i.e., $\eta \sim \frac{\gamma}{2}e^{-\gamma|x|}$, where γ is the laplace coefficient and $E[\eta] = 0$) or Uniform distribution (i.e., $\eta \sim \frac{1}{2r}$, where r denotes the half range of the distribution, the entire being $[-r:r]$ leading to $E[\eta] = 0$). Different levels of corruption are examined for these cases as well.

Four objective criteria have been evaluated for each pair of noisy and filtered images, namely the SNR, the $PSNR$, the MAE and the MSE, defined by

$$S\hat{N}R = 10\log_{10}\frac{\hat{\sigma}_d^2}{\hat{\sigma}_\varepsilon^2} \qquad PS\hat{N}R = 10\log_{10}\frac{255^2}{\hat{\sigma}_\varepsilon^2} \tag{23}$$

$$M\hat{A}E = \frac{1}{\mathcal{KL}}\sum_{k=1}^{\mathcal{K}}\sum_{l=1}^{\mathcal{L}}|\varepsilon(k,l)| \qquad M\hat{S}E = \frac{1}{\mathcal{KL}}\sum_{k=1}^{\mathcal{K}}\sum_{l=1}^{\mathcal{L}}\varepsilon^2(k,l) \tag{24}$$

In the above equations, $\varepsilon(k,l) = y(k,l) - d(k,l)$ is the output noise, where $d(k,l)$ is the noise-free image and $y(k,l)$ denotes the filtered image. $\hat{\sigma}_d^2$ is the variance of the noise-free image and $\hat{\sigma}_\varepsilon^2$ is the variance of the output noise. $\mathcal{K}, \mathcal{L}$ correspond to the number of rows and columns, respectively, of the processed part of the test image in question.

To begin with let us describe the experimental setup. Two different noise-free images have been chosen, namely, the images "Airfield" and "Bridge". Initially, both of them have been corrupted by adding white i.i.d. noise obeying the pdf of a Gaussian mixture given by (22). For the adaptive LMS order statistic filters, a preprocessing phase (i.e., a training phase) has been employed aiming at deriving the filter coefficients following the procedure outlined subsequently. A noise-free training image and a corrupted one by noise having the same distribution to the one observed in the test image to be filtered are used. The filter structure that is used to filter the test image is also used in the training phase. The test image is filtered by using the coefficients determined when the training phase has led to convergence. These steady state coefficients can be obtained either by averaging the filter coefficients found in the last row of the training image or by choosing the coefficients found in the last image pixel filtered in the training phase. Although a noise-free version of the training image (e.g. a TV logo) still is needed, there is no such need for a noise-free version of the test image.

Following the above-described procedure, firstly, a corrupted version of image "Bridge" has been employed in the training phase, that aims at determining the LMS order statistic filter coefficients at convergence. A value of $\mu = 10^{-7}$ has been employed in the filters' coefficient updating equations 8 and 13 during training. A corrupted version of image "Airfield" by the same kind of noise has been used as a test image. The coefficients determined at the end of the training phase are applied to the test image for noise smoothing. The opposite situation has also been tested. That is, a corrupted version of image "Airfield" has been employed in the training phase and a corrupted version of image "Bridge" by the same kind of noise has been used as a test image. A window size of 3×3 has been used in both training and testing phases. Images "Airfield" and "Bridge" both of dimensions 512×512 are shown in Figures 2a and 2b, respectively. Three noise cases are studied in more detail:

Case 1: The test images are corrupted by noise having a contaminated Gaussian distribution (22) with values for $\lambda = 0.1$ and $\sigma_\nu^2 \simeq 586.5$. This leads to a high corruption of the original image by mixed impulsive and Gaussian noise. The corrupted images are called "Airm0s3l01" and "Brim0s3l01" and are shown in Figures 3a and 4a, respectively.

Case 2: λ is chosen to be equal to 1.0 which implies that only pure Gaussian noise is present in the noisy image. A value close to 474.0 has been used for σ_ν^2. The resulting images are named "Airm0s9l10" and "Brim0s9l10" and are depicted in Figures 5a and 6a, respectively.

Case 3: Impulsive noise of 10% positive and 10% negative impulses of constant value has been added to the test images to examine the robustness of the proposed filters in pure impulsive noise. The final images "Airfield.i" and "Bridge.i" are shown in Figures 7a and 8a, respectively.

The quantitative figures of merit for the three cases are tabulated in Table 2 when the noisy versions of images "Airfield" and "Bridge" are filtered. The respective figures of merit for the SAM and the median filter of dimensions 3×3 are included in Table 2 for comparison purposes. Table 3 contains the values of the $MSAM/SAM$ filter parameters and thresholds (p_p,p_n,α,β) employed in the simulation process. The inspection of Table 2 leads to the following conclusions concerning the performance of the adaptive order statistic filters under study with respect to SNR, $PSNR$, MAE and MSE values:

- For low SNR and very impulsive images, the morphological SAM filter attains the best performance. The Ll-filter is the second best.

- For images corrupted by Gaussian noise, the adaptive Ll filter attains the best performance. The $MSAM$ filter is found to be the second best filter.
- For pure impulsive noise $MSAM$ outperforms the other filters in terms of SNR, $PSNR$ and MSE values whereas SAM attains the smallest MAE.
- The location invariant L-filter is not far behind in all cases.

It is worth noting that all filters under study outperform the median filter.

The filtered images by the adaptive order statistic filters under study are provided in order to evaluate the filter performance with respect to the achieved visual quality. Figures 3b-d show the outputs of the location invariant L-, adaptive Ll- and morphological SAM filters, respectively, when the noisy image "Airm0s3l01" is filtered. For comparison purposes, the outputs of the SAM and the median filter of dimensions 3×3 are shown in Figures 3e-f, respectively. Figures 4b-f show the outputs of the above-mentioned filters in the same order when the noisy image "Brim0s3l01" is filtered. The outputs of the filters under study when image "Airm0s9l10" is filtered are depicted in Figures 5b-f and the outputs of the same filters when image "Brim0s9l10" is filtered are shown in Figures 6b-f respectively. Figures 7b-f and 8b-f demonstrate the outputs of the filters under study when images "Airfield.i" and "Bridge.i" are filtered, respectively. By observing these figures, it is found that the $MSAM$-filtered image is of better perceived visual quality than all the other filtered images. Edges are preserved and not blurred, which is not the case with the other filters under study. Mixed impulsive and Gaussian noise is smoothed very well, an untrue fact for the other filters. Moreover, the removal of impulses and Gaussian noise in homogeneous image regions is more successful when $MSAM$ is used in all corruption cases examined. The superior $MSAM$ filter performance is attributed to three factors:

(i) the use of large window sizes in homogeneous regions,

(ii) the anisotropic window increment that allows a higher noise suppression close to edges than the one achieved by the standard SAM filter, and

(iii) the improved impulse detection mechanism that is now employed.

In order to justify the above remarks, Figures 9a-b and 10a-b are included. Figures 9a-b show the total number of pixels in the "optimal" windows employed during filtering of a small part of "Airm0s9l10" by the SAM and $MSAM$ filters, respectively. It is seen that $MSAM$ yields larger windows nearby edges than SAM resulting in better noise smoothing near edges. Figures 10a-b illustrate the value of $b(n)$ appropriately scaled after filtering the same noisy image. A value close to 1.0 indicates edge region while a value close to 0.0 a flat one. SAM edge detection produces thicker edges which implies less filtering in those areas. Furthermore, when impulsive noise is present, pixels in flat regions are mistakenly assigned as edge pixels by SAM due to its reduced impulse detection capabilities.

It is worth noting that such a good performance of $MSAM$ is obtained without any reference image or training procedure and in a single pass. Furthermore, the poor performance of median filters in presence of Gaussian noise is diminished. It is remarkable that the visual perception of the images filtered by the three filters discussed in the paper is superior to that of the median filtered image.

Performance results for a wider range of corruption levels in case of contaminated Gaussian noise presence, as well as for a wide range of corruption levels when additive Laplacian or Uniformly distributed noise is added, are tabulated in Table 4 for the SAM and $MSAM$ filters, in order to justify the improved performance of the $MSAM$ filter. Achieved SNR and MSE values are illustrated for corrupted versions of image "Airfield". The inspection of this Table leads to similar conclusions analysed in the above. In almost all cases, the $MSAM$ filter outperforms the SAM and in the remaining ones their performance is almost identical. However, the perceived visual quality of the filtered images by $MSAM$ is much more superior than the one achieved by SAM.

Remaining noise in flat regions and non-removed impulses are two significant drawbacks of the latter. It should be noted that a minor disadvantage of the $MSAM$ filter is its inability to preserve very small details and thin lines.

4-B Complexity evaluation

The main disadvantage of the adaptive Ll-filter is its high memory requirements and increased computational complexity due to the large number of filter coefficients that need to be adapted in the training phase and stored in both phases. Its complexity and storage needs are analogous to N^2, where N is the number of pixels in the filter window. Consequently, if a larger window size is used, the filter's complexity and memory needs increase significantly. The adaptive location-invariant L-filter has the advantages of simple structure, limited memory requirements and small computational complexity. With respect to computational complexity and memory requirements, the $MSAM$ filter lies in-between, if a large window size (e.g. 7×7) is used by the LMS adaptive filters. Table 5 contains the execution time spent by the three filters under study in order to filter images "Airm0s3l01", "Airm0s9l10" and "Airfield.i". Simulations have been performed on an IRIS Silicon Graphics INDY machine with a MIPS R4400 Processor of 200MHz. The LMS adaptive filter execution time is estimated for two different window sizes: 3×3 and 7×7. The latter is considered in this analysis so that their execution time is comparable with that of the $MSAM$ filter, which employs variable filter window sizes starting from 3×3 and reaching up to 11×11. The respective values for the SAM filter are also shown for comparison purposes.

Examining now the window adaptation processes, the anisotropic window adaptation procedure of $MSAM$ requires more computational effort than the isotropic window adaptation used in the standard SAM due to erosion and dilation computations, as is also shown in Table 5.

5 Conclusions

The application of adaptive order statistics filters in noise removal from corrupted images has been studied in this paper. Three adaptive order statistics filters have been considered, namely, the adaptive location-invariant L-, the adaptive Ll- and the morphological SAM filters. The adaptive location-invariant L-filter and the adaptive Ll-filter were proposed by the authors in [8, 10, 11] in the past. The morphological signal-adaptive median filter is a novel extension of the signal-adaptive filter [18] proposed in this paper. The $MSAM$ filter requires neither a priori knowledge of the noise-free version of the corrupted image nor a training phase. Therefore, the $MSAM$ filter is highly appropriate for image filtering. The noise suppression capabilities of the proposed $MSAM$ filter have been compared to the ones of the LMS location-invariant L- and LMS Ll-filter whose performance has been used as a baseline in the comparative study. It has been demonstrated that the highest gain in performance both quantitatively and qualitatively is obtained when low SNR and very impulsive images are encountered. Although $MSAM$ has a higher computational complexity than the SAM filter, it is still attractive because its computational complexity combined with its memory requirements lie between those of its competitors (i.e., the LMS location-invariant L-filter and the LMS Ll-filter).

References

[1] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, N.J, 1986.

[2] I. Pitas and A.N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*, Kluwer Academic, Dordrecht, Holland, 1990.

[3] I. Pitas and A.N. Venetsanopoulos, "Order statistics in digital image processing", *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1893–1921, December 1992.

[4] T.S. Huang A.C. Bovik and D.C. Munson, "A generalization of median filtering using linear combinations of order statistics", *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 31, no. 6, pp. 1342–1349, December 1983.

[5] F. Palmieri and C.G. Boncelet, "Ll-filters - A new filter class of order statistic filters", *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 37, no. 5, pp. 691–701, March 1989.

[6] K.E. Barner and G.R. Arce, "Permutation filters - A class of nonlinear filters based on set permutations", *IEEE Trans. on Signal Processing*, vol. 42, no. 4, pp. 782–798, April 1994.

[7] I. Pitas and A.N. Venetsanopoulos, "Adaptive filters based on order statistics", *IEEE Trans. on Signal Processing*, vol. 39, no. 2, pp. 518–522, February 1991.

[8] C. Kotropoulos and I. Pitas, "Constrained adaptive LMS L-filters", *Signal Processing*, vol. 26, no. 3, pp. 335–358, 1992.

[9] G.A. Williamson and P.M. Clarkson, "On signal recovery with adaptive order statistic filters", *IEEE Trans. on Signal Processing*, vol. 40, no. 10, pp. 2622–2626, October 1992.

[10] C. Kotropoulos and I. Pitas, "Adaptive LMS L-filters for noise suppression in images", *IEEE Trans. on Image Processing*, vol. 5, no. 12, pp. 1596–1609, December 1996.

[11] I. Pitas and S. Vougioukas, “LMS order statistic filter adaptation by backpropagation”, *Signal Processing*, vol. 25, pp. 319–335, 1991.

[12] Y.-T. Kim and G.R. Arce, “Permutation filter lattices: A general order statistic filtering framework”, *IEEE Trans. on Signal Processing*, vol. 42, no. 9, pp. 2227–2241, September 1994.

[13] G.R. Arce and M. Tian, “Order statistic filter banks”, *IEEE Trans. on Image Processing*, vol. 5, no. 6, pp. 827–837, June 1996.

[14] C. Kotropoulos and I. Pitas, *Control and Dynamic Systems*, vol. 67, chapter Adaptive nonlinear filters for digital signal/image processing, pp. 263–318, Academic Press, San Diego, CA, 1994.

[15] J.S. Lee, “Digital image enhancement and filtering by use of local statistics”, *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 2, pp. 165–168, March 1980.

[16] S.K. Mitra M. Lightstone, E. Abreu and K. Arakawa, “A new filtering approach for the removal of impulse noise from highly corrupted images”, *IEEE Trans. on Image Processing*, vol. 5, no. 6, pp. 1012–1025, June 1996.

[17] I. Pitas, “Fast algorithm for running ordering and max/min selection”, *IEEE Trans. on Circuits and Systems*, vol. 36, no. 6, pp. 795–804, June 1989.

[18] R. Bernstein, “Adaptive nonlinear filters for simultaneous removal of different kinds of noise in images”, *IEEE Trans. on Circuits and Systems*, vol. 34, no. 11, pp. 1275–1291, November 1987.

[19] M. Gabbouj T. Sun and Y. Neuvo, “Center weighted median filters: Some properties and application in image processing”, *Signal Processing*, vol. 35, no. 3, pp. 213–229, February 1994.

[20] I.D. Svalbe, "The geometry of basis sets for morphologic closing", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 12, pp. 1214–1224, December 1991.

[21] R. Sucher, "A recursive nonlinear filter for the removal of impulse noise", in *1995 IEEE Int. Conf. on Image Processing*, 1995, pp. 183–186.

[22] X.Z. Sun and A.N. Venetsanopoulos, "Adaptive schemes for noise filtering and edge detection by use of local statistics", *IEEE Trans. on Circuits and Systems*, vol. 35, no. 1, pp. 57–69, January 1988.

[23] G.J. Yang T.S. Huang and G.Y. Tang, "A fast two-dimensional median filtering algorithm", *IEEE Trans on Acoustics, Speech and Signal Processing*, vol. 27, no. 1, pp. 13–18, February 1979.

[24] J.T. Astola and T.G. Campbell, "On computation of the running median", *IEEE Trans on Acoustics, Speech and Signal Processing*, vol. 37, no. 4, pp. 572–574, April 1989.

[25] M. Gabbouj and I. Tabus, "TUT noisy image database", Tech. Rep. ISBN 951-722-281-5, Signal Processing Lab., Tampere Univ. of Technology, Tampere, Finland, December 1994.

List of Tables

List of Figures

Table 1: *MSAM* and *SAM* filter complexity analysis in terms of total number of additions/subtractions, multiplications/divisions and comparisons required for the estimation of an output sample.

Process	*MSAM* Complexity	
	adds/subs	mults/divs
"Windowed" Histogram [18]	$N_I + \sum_{i=1}^{S_D}(n_{S_D,i}) + 2\sum_{j=1}^{S_E}(n_{S_E,j}) + S_D + 2S_E + 4S_T + 2$	$4S_T$
Impulse Detection	6	2
$b(n)$ $(\beta = 1.0)$	$2(N_I + \sum_{i=1}^{S_D}(n_{S_D,i}) + 2\sum_{j=1}^{S_E}(n_{S_E,j})) + 4S_T$	$N_I + \sum_{i=1}^{S_D}(n_{S_D,i}) + 2\sum_{j=1}^{S_E}(n_{S_E,j}) + S_D + 2S_E + 5S_T$
Output	2	1
	comparisons	
Modified Median [23]	$2\lceil\sqrt{N_I - \Delta h_{min,I} - \Delta h_{max,I} - 1}\rceil + 2\lceil\sqrt{N_F - \Delta h_{min,F} - \Delta h_{max,F} - 1}\rceil$	

Process	*SAM* Complexity	
	adds/subs	mults/divs
"Windowed" Histogram [18]	$N_F + 5S_{T,SAM} + 2$	$4S_{T,SAM}$
Impulse Detection	incl. below	incl. below
$b(n)$ $(\beta = 1.0)$	$2N_F + 7S_{T,SAM}$	$N_F + 7S_{T,SAM}$
Output	2	1
	comparisons	
Modified Median [23]	$\sum_{i=1}^{S_{T,SAM}} 2\lceil\sqrt{N_i - \Delta h_{min,i} - \Delta h_{max,i} - 1}\rceil$	

Table 2: Figures of merit when corrupted versions of both images “Airfield” and “Bridge” with varying noise parameters are filtered. The best results are shown in bold.

Filter	Airm0s3l01				Brim0s3l01			
	$\widehat{SNR}$	$\widehat{PSNR}$	$\widehat{MAE}$	$\widehat{MSE}$	$\widehat{SNR}$	$\widehat{PSNR}$	$\widehat{MAE}$	$\widehat{MSE}$
Initial	3.034	15.608	26.467	1788.423	2.998	16.363	23.445	1503.544
MSAM	**11.721**	**24.295**	**11.005**	**242.177**	11.213	24.579	**10.984**	226.831
Ll	11.650	24.224	12.159	254.632	**11.458**	**24.824**	11.097	**217.289**
L	11.553	24.127	11.840	254.127	11.155	24.521	11.436	235.737
SAM	9.272	21.846	13.157	425.840	9.551	22.917	12.106	332.790
Median	11.188	23.762	12.273	273.558	10.817	24.182	11.788	248.402
	Airm0s9l10				Brim0s9l10			
	$\widehat{SNR}$	$\widehat{PSNR}$	$\widehat{MAE}$	$\widehat{MSE}$	$\widehat{SNR}$	$\widehat{PSNR}$	$\widehat{MAE}$	$\widehat{MSE}$
Initial	9.011	21.586	16.829	451.380	9.017	22.382	15.467	375.966
MSAM	12.796	25.370	10.276	188.930	12.411	25.777	9.930	172.016
Ll	**13.367**	**25.942**	**9.901**	**165.538**	**13.164**	**26.530**	**9.259**	**144.608**
L	12.664	25.238	10.433	197.512	11.880	25.245	10.440	196.461
SAM	12.466	25.041	10.764	203.791	11.934	25.300	10.345	192.016
Median	12.384	24.958	10.654	207.795	11.517	24.882	10.859	211.424
	Airfield.i				Bridge.i			
	$\widehat{SNR}$	$\widehat{PSNR}$	$\widehat{MAE}$	$\widehat{MSE}$	$\widehat{SNR}$	$\widehat{PSNR}$	$\widehat{MAE}$	$\widehat{MSE}$
Initial	-0.026	12.548	23.040	3628.188	-0.688	12.678	23.085	3520.368
MSAM	**14.468**	**27.043**	4.599	**129.390**	**14.940**	**28.306**	3.101	**96.489**
Ll	12.758	25.332	8.746	196.527	12.459	25.825	7.737	170.873
L	12.324	24.898	8.750	218.376	11.760	25.125	9.188	109.496
SAM	13.407	25.981	**3.082**	164.407	14.607	27.973	**2.874**	103.810
Median	12.580	25.154	7.690	198.527	11.645	25.010	8.070	205.147

Table 3: $MSAM/SAM$ filter parameter and threshold values.

Noisy Image	p_p	p_n	b_t	α	β
Airm0s3l01	0.0612	0.0369	0.4	0.3	1.0
Brim0s3l01	0.0337	0.0384	0.4	0.3	1.0
Airm0s9l10	0.0	0.0	0.5	0.6	1.0
Brim0s9l10	0.0	0.0	0.45	0.7	1.0
Airfield.i	0.1156	0.0910	0.3	1.0	1.0
Bridge.i	0.0918	0.0948	0.4	1.0	1.0

Table 4: Comparison of SAM and $MSAM$ filter performances for a wide range of corruption cases by Contaminated Gaussian noise. Additional performance results are incorporated for the cases of additive Laplacian and Uniform noise distributions.

Contaminated Gaussian							
Noisy Image	λ	$\hat{\sigma}_\eta^2$	Initial	$MSAM$		SAM	
			SNR	SNR	MSE	SNR	MSE
Airm0s9l01	0.1	451.122	9.013	14.973	114.421	11.369	262.534
Airm0s15l01	0.1	113.162	15.019	17.500	63.924	14.114	139.494
Airm0s3l02	0.2	1807.241	2.986	10.763	302.290	7.729	608.292
Airm0s9l02	0.2	449.976	9.024	14.125	139.099	11.178	274.443
Airm0s15l02	0.2	113.616	15.002	17.016	71.477	14.288	133.938
Airm0s3l10	1.0	1790.666	3.026	8.818	472.208	8.164	548.732
Airm0s15l10	1.0	113.371	15.011	17.548	63.223	17.638	62.065

Laplacian							
Noisy Image	γ	$\hat{\sigma}_\eta^2$	Initial	$MSAM$		SAM	
			SNR	SNR	MSE	SNR	MSE
Airm0s20l4	4.0	31.884	20.521	21.714	24.443	21.731	24.130
Airm0s14l8	8.0	124.735	14.596	17.082	70.760	17.007	71.608
Airm0s11l12	12.0	274.780	11.166	14.481	128.531	12.535	201.188
Airm0s9l16	16.0	475.530	8.785	12.987	181.396	10.797	300.168

Uniform							
Noisy Image	r	$\hat{\sigma}_\eta^2$	Initial	$MSAM$		SAM	
			SNR	SNR	MSE	SNR	MSE
Airm0s18r25	25.0	51.031	18.478	20.199	34.623	20.227	34.118
Airm0s12r50	50.0	203.568	12.470	15.638	98.591	15.518	100.924
Airm0s9r75	75.0	453.644	8.990	13.217	172.087	12.930	183.172

Table 5: Execution time of filters under study. Additional information about isotropic - anisotropic window adaptation execution times is included.

Filter	Airm0s3l01		Airm0s9l10		Airfield.i	
	Total	Window Ad	Total	Window Ad	Total	Window Ad
$MSAM$	55.09s	33.79s	55.03s	40.73s	40.27s	18.07s
SAM	24.43s	19.06s	18.4s	13.03s	15.74s	10.04s
Ll 3 × 3	13.07s	-	13.06s	-	12.97s	-
Ll 7 × 7	65.62s	-	64.01s	-	64.99s	-
L 3 × 3	3.76s	-	4.22s	-	3.82s	-
L 7 × 7	45.61s	-	45.42s	-	45.09	-

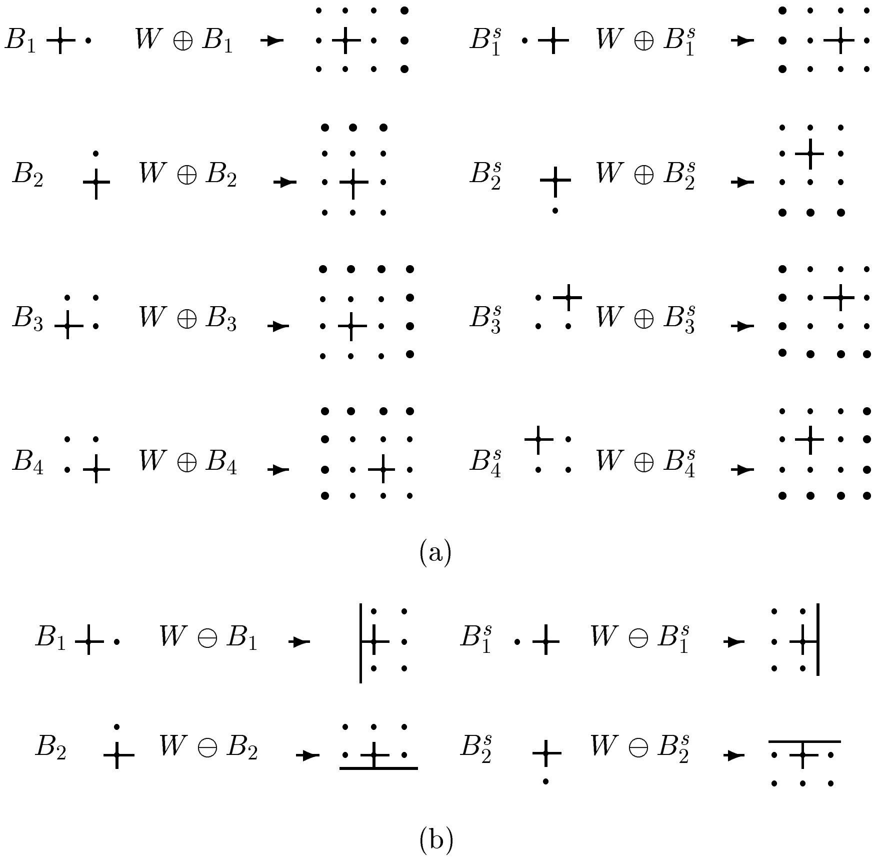


Figure 1: Structuring Sets: (a) Window increment through dilation. (b) Window decrement through erosion.

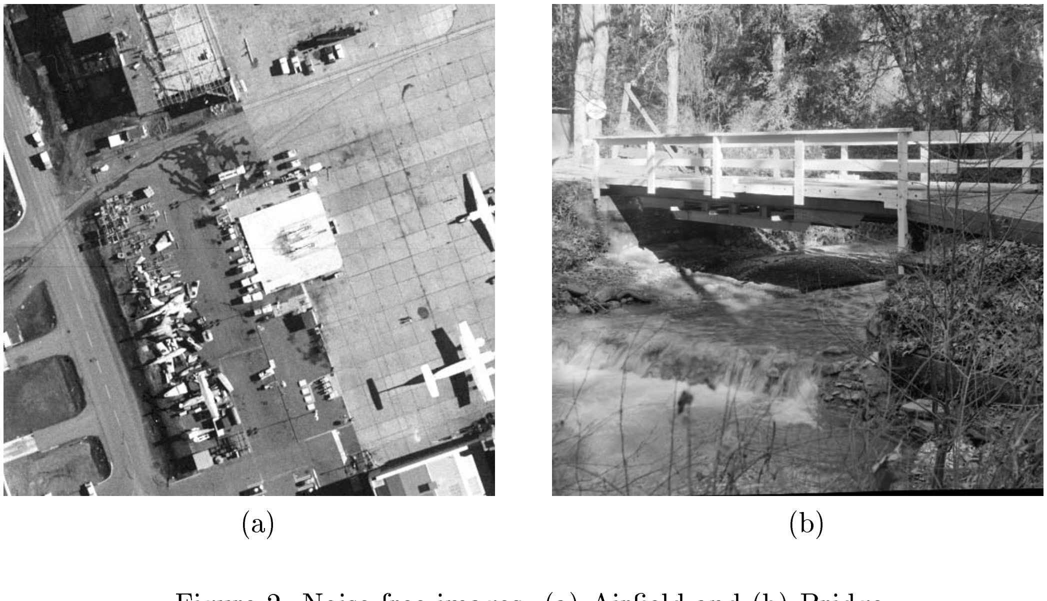
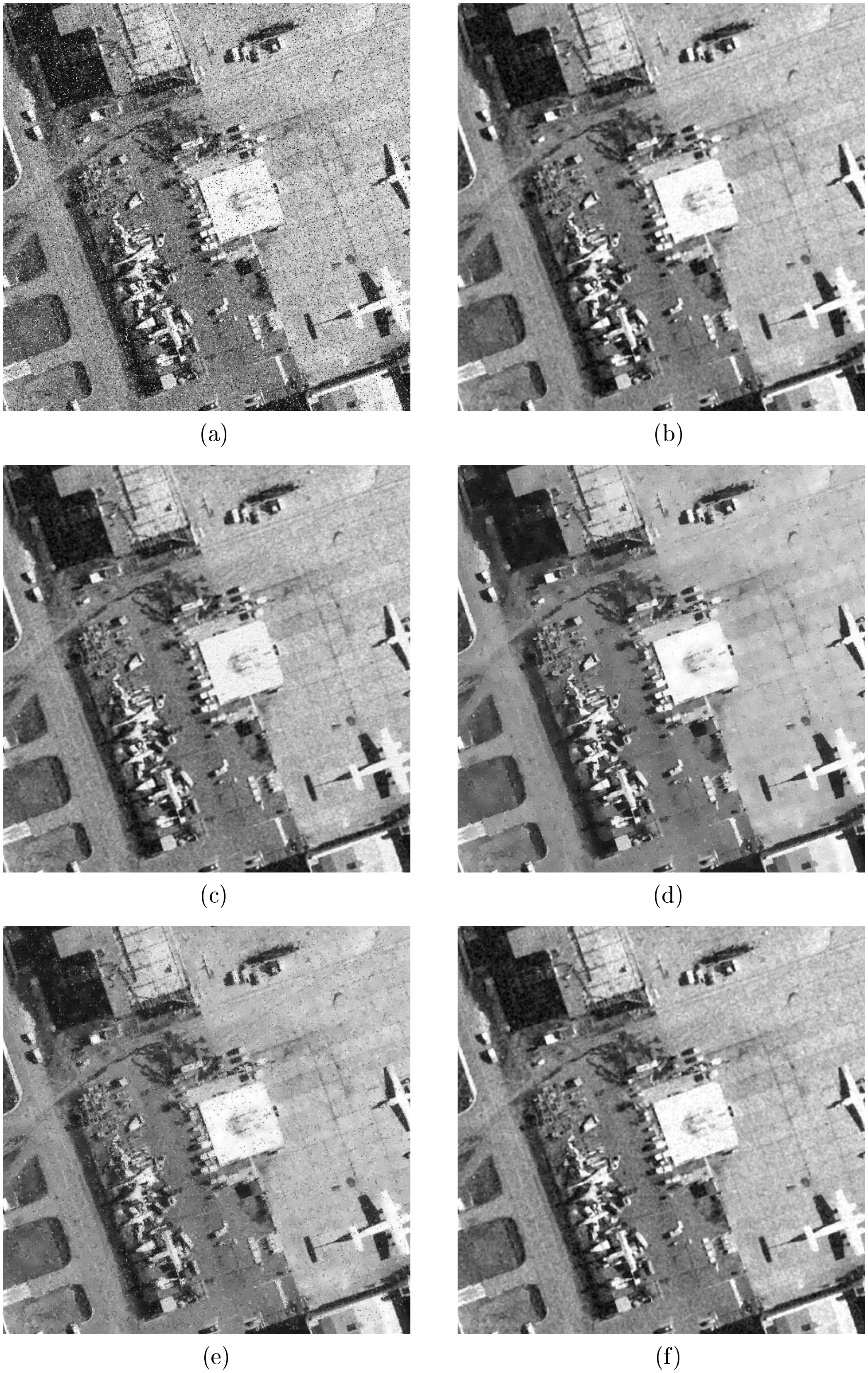


Figure 2: Noise-free images: (a) Airfield and (b) Bridge.

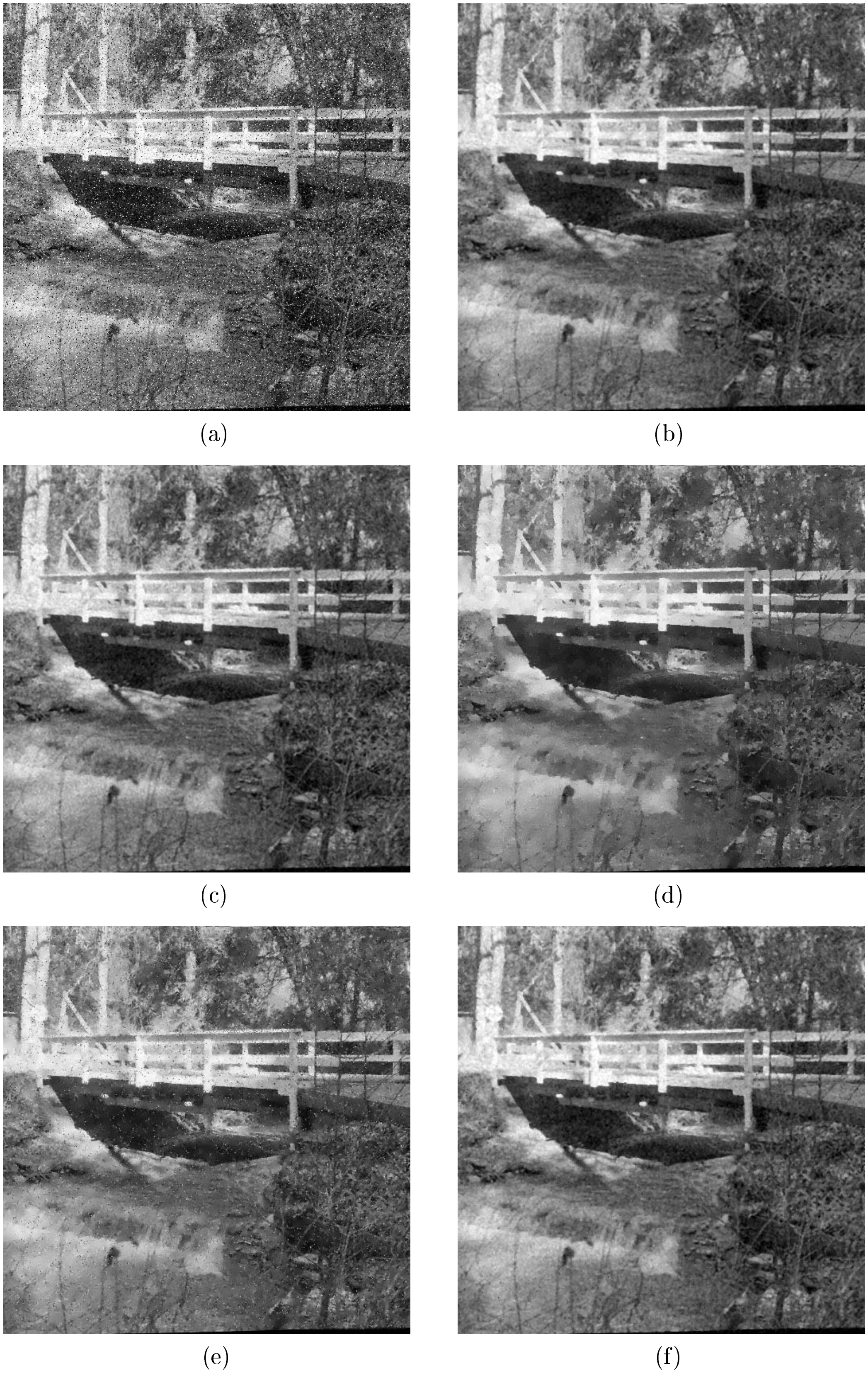


(a) (b)

(c) (d)

(e) (f)

Figure 3: (a) Noisy image: Airm0s3l01. Filtered Airm0s3l01 by (b) the location invariant L filter, (c) the adaptive Ll filter, (d) the morphological SAM filter, (e) the SAM filter, (f) the median filter using a 3×3 mask.



(a)

(b)

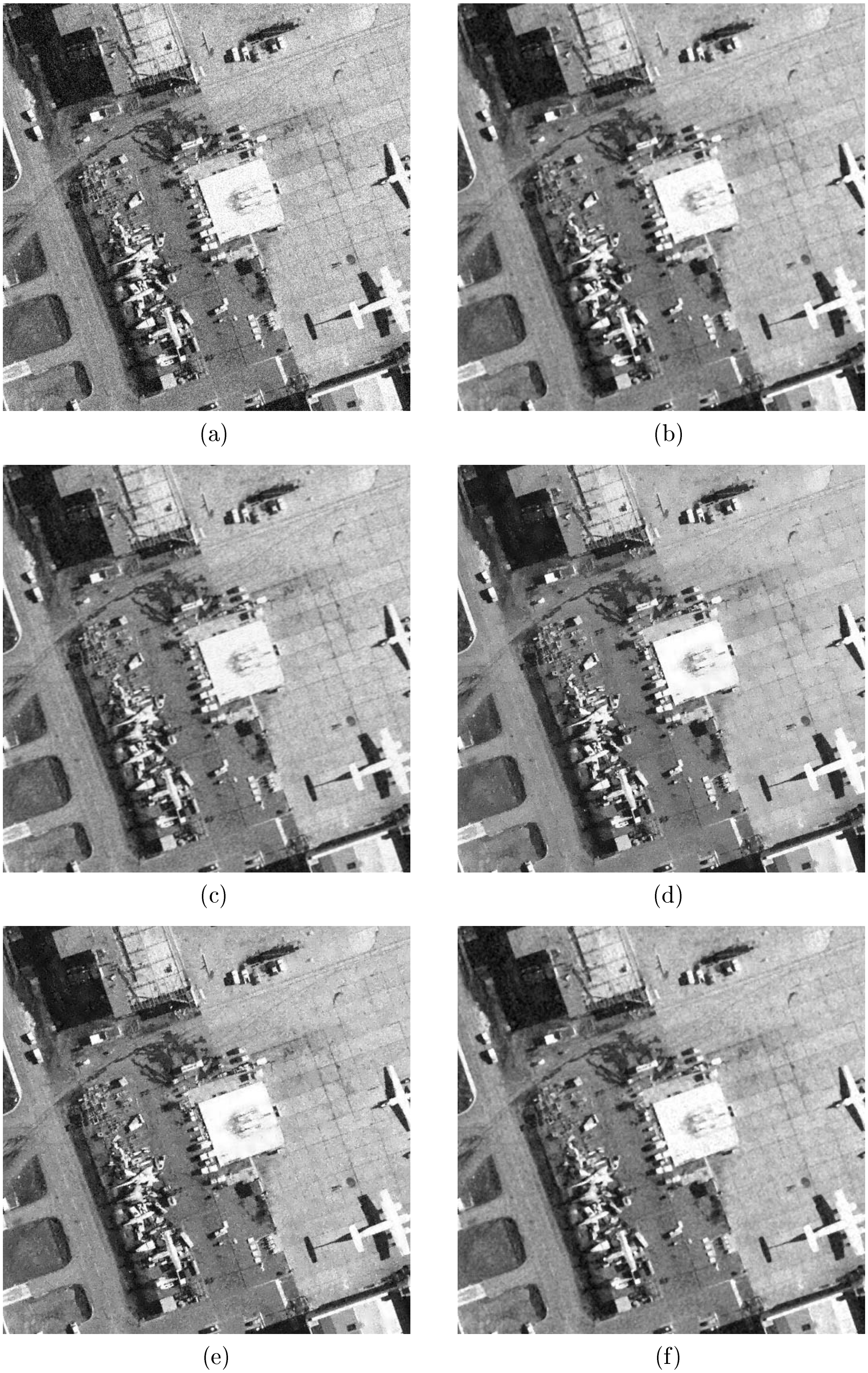
(c)

(d)

(e)

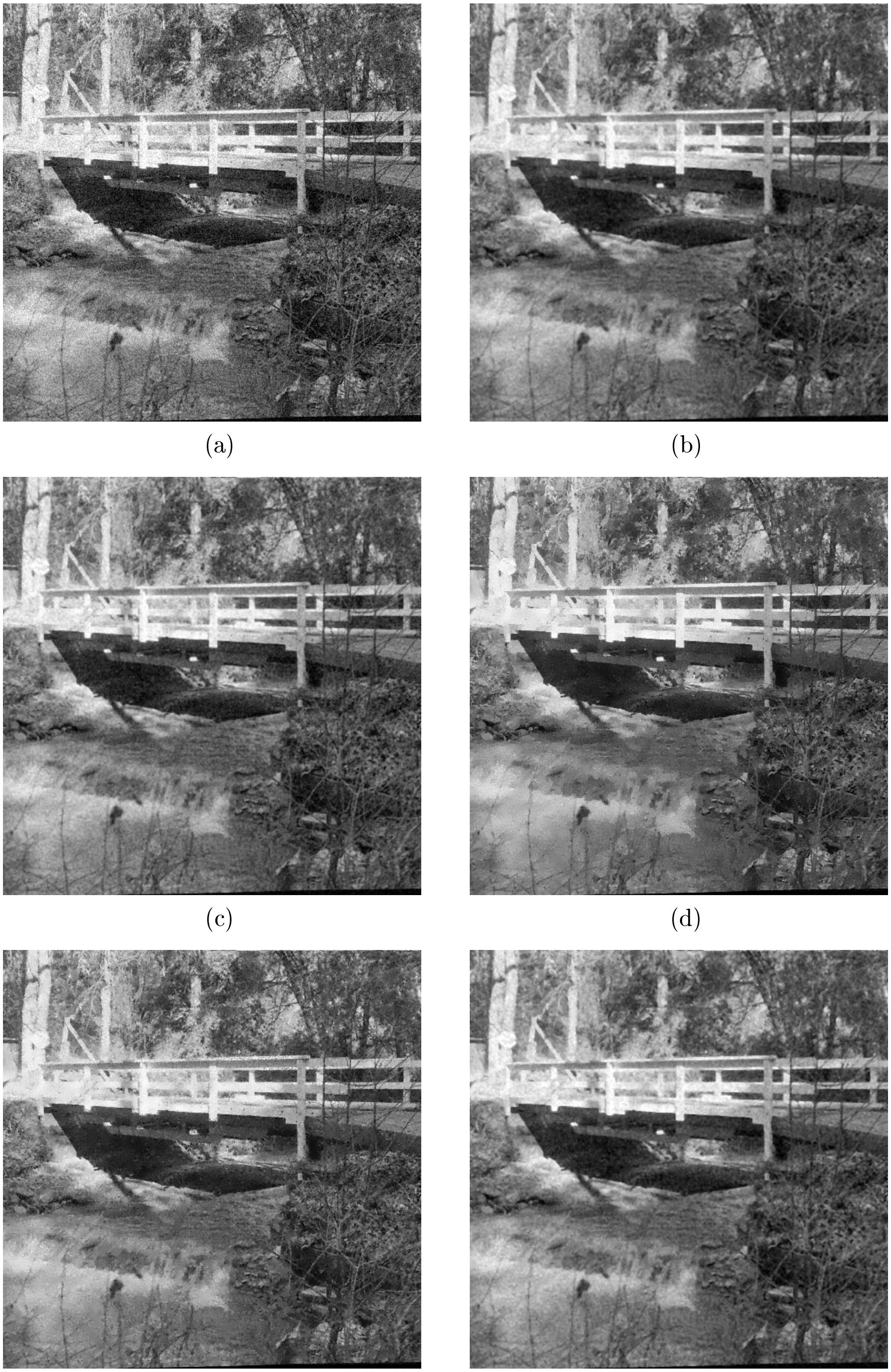
(f)

Figure 4: (a) Noisy image: Brim0s3l01. Filtered Brim0s3l01 by (b) the location invariant L filter, (c) the adaptive Ll filter, (d) the morphological SAM filter, (e) the SAM filter, (f) the median filter using a 3×3 mask.



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

Figure 5: (a) Noisy image: Airm0s9l10. Filtered Airm0s9l10 by (b) the location invariant L filter, (c) the adaptive Ll filter, (d) the morphological SAM filter, (e) the SAM filter, (f) the median filter using a 3×3 mask.



(a)

(b)

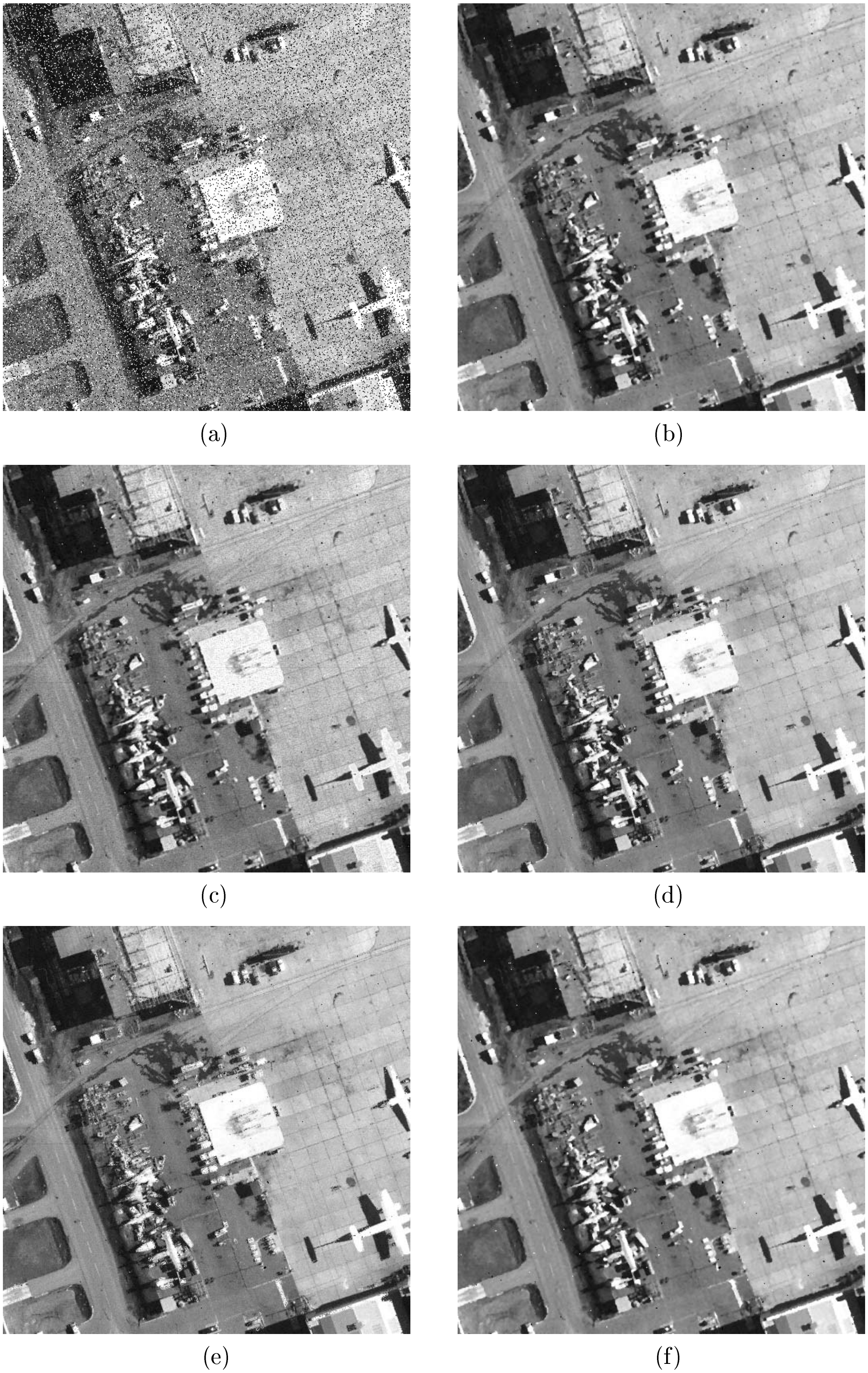
(c)

(d)

(e)

(f)

Figure 6: (a) Noisy image: Brim0s9l10. Filtered Brim0s9l10 by (b) the location invariant L filter, (c) the adaptive Ll filter, (d) the morphological SAM filter, (e) the SAM filter, (f) the median filter using a 3×3 mask.

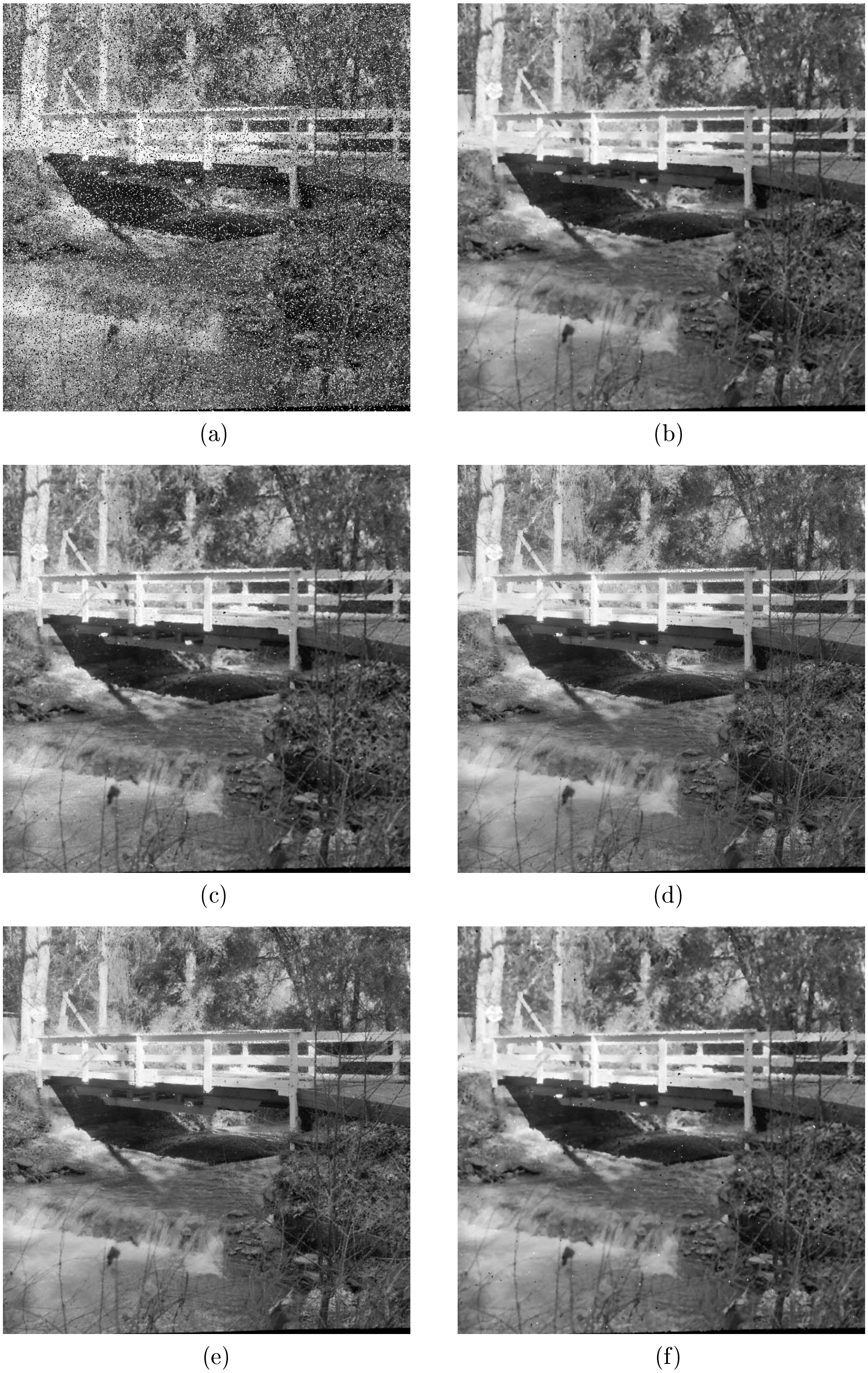


(a) (b)

(c) (d)

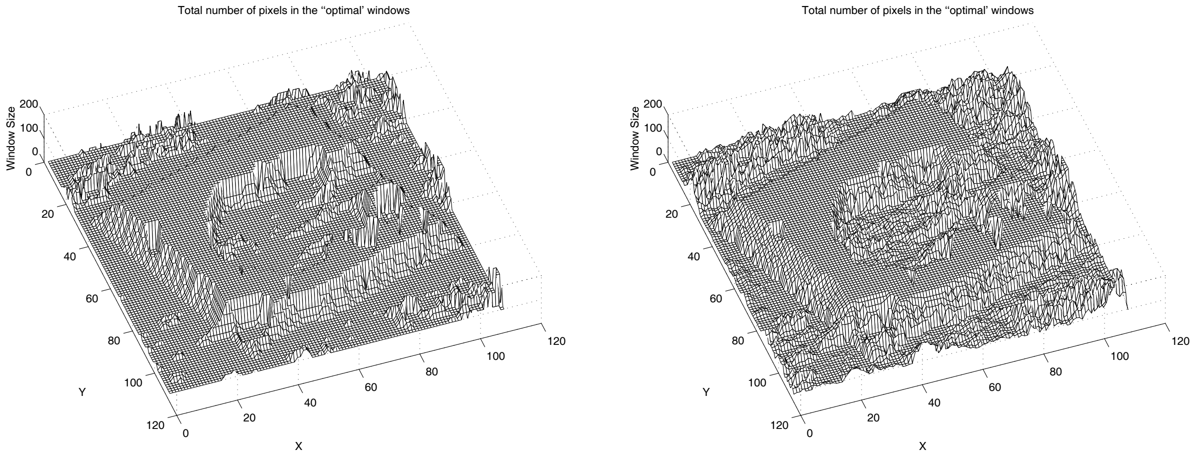
(e) (f)

Figure 7: (a) Noisy image: Airfield.i. Filtered Airfield.i by (b) the location invariant L filter, (c) the adaptive Ll filter, (d) the morphological SAM filter, (e) the SAM filter, (f) the median filter using a 3×3 mask.



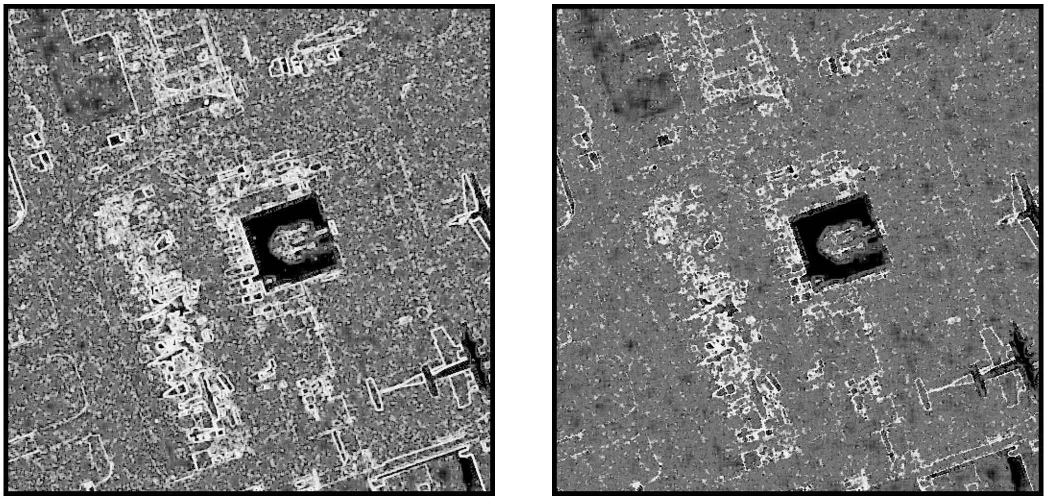
(a) (b) (c) (d) (e) (f)

Figure 8: (a) Noisy image: Bridge.i. Filtered Bridge.i by (b) the location invariant L filter, (c) the adaptive Ll filter, (d) the morphological SAM filter, (e) the SAM filter, (f) the median filter using a 3×3 mask.



(a) (b)

Figure 9: Total number of pixels in "optimal windows" employed when a part of Airm0s9l10 is filtered by (a) the SAM and (b) the $MSAM$ filter.



(a) (b)

Figure 10: Estimated values of $b(n)$ after filtering Airm0s9l10 by (a) the SAM and (b) the $MSAM$ filter.