Detail-Preserving Image Information Restoration guided by SVM based noise mapping

P. Pankajakshan [a,*], V. Kumar [b]

[a] *Texas A&M University, Department of Electrical Engineering, College Station, TX 77843, USA*

[b] *Indian Institute of Technology Roorkee, Department of Electrical Engineering, Roorkee-247 667, Uttaranchal, India*

Abstract

In this paper, we propose a new method to solve the problem of impulsive noise reduction in images. Non-linear filter like the median filter (MF) is useful for reducing random noise and periodical patterns, but direct median filtering have undesirable side effects such as smoothening of noise free regions, which results in loss of image detail and distortion of signal. Impulse noise is suppressed by selectively filtering the contaminated signal regions only, thus minimizing distortion of clean passages and loss of high frequencies. In the first phase, Support Vector Machines (SVM) are used to segment the set of pixels N that are likely to be contaminated by the mixed impulses. In the second phase, the image is restored by employing a combination of the best neighborhood match filter (BNM) and the modified multi-shell median filter (MMMF) to these segmented regions. This method combines the effectiveness of the Best Neighborhood Matching (BNM) filter in suppression of the noise components while adapting itself to the local image structures, and the edge and finer image detail preserving characteristics of the MMMF. To support our proposed method, numerical results are also provided, which indicate that the filter is extremely useful for preserving edges or monotonic changes in trend, while eliminating impulses of short duration.

Key words: Impulse noise, noise suppression, median filter (MF), kernel function, support vector machines (SVM)

* Corresponding author. Texas A&M University, Department of Electrical Engineering, College Station, TX 77843, USA

Email addresses: praveenpankaj@ieee.org, vinodfee@iitr.ernet.in (V. Kumar).

1 Introduction

Impulse noise is a special type of noise that can come from a variety of sources. Standard broadcast TV signals or satellite images are frequently contaminated with this noise, arising from vehicle-engine, electrical-appliance or atmospheric disturbances. In some other applications it can be caused by transmission errors, malfunctioning pixels in camera sensors, and faulty memory locations in hardware. "Salt and Pepper" noise can also be generated from bit errors in the data stream and they occur as isolated impulses. The image is corrupted with individual noisy pixels whose intensity significantly differs from the neighboring pixels. By noise, we refer to stochastic variations rather than deterministic distortions. Modern technology has made it possible to reduce the noise levels to almost negligible levels.

In many practical situations, the probability of spikes is low and two or more neighboring pixels are very seldom corrupted by impulsive noise. In other words, the spikes possess an approximately spatially invariant characteristic. Many efficient and robust filtering algorithms have been already proposed to remove spikes that fulfill the aforementioned model assumptions. However, these assumptions are not valid in some practical situations. For example, interference may occur when the data is transferred using analog signal communication channel. This interference can be long term and so intensive that it corrupts several consecutive image pixels in one or more columns following each other (we assume that the images are transferred as rows). Such situations may happen if the receiver input and circuitry are not well protected against intensive interference, or if there are some electromagnetic wave irradiation sources in the neighborhood that are overlaps with the frequency band of the channel waveband. There are different amounts of horizontal impulse bursts that appear in these images. This kind of bursts, appearing as line-type noise considerably decrease the image quality.

Although it is widely known that non-linear filter like the median filter (MF) is useful for reducing random noise and periodical patterns. However, direct median filtering have undesirable side effects such as smoothening of noise free regions, which results in loss of image detail and distortion of signal. The problem of noise elimination in an image has a companion limitation which is the distortion of image edges. The requirement of maximal preservation of edges is especially important for images, corrupted by impulsive noise with a low corruption rate. To avoid smoothening of the image during filtering, all noisy pixels must be detected first, and only then, these detected pixels must be corrected. Other nonlinear filters, proposed for impulsive noise reduction (for example, rank-order, weighted, relaxed median filters) preserve image edges, but in general, the results are not good enough. A good way to solve the preservation problem is noise detection. If the noisy pixels are detected and they are a priori known before filtering, then the filter can be applied only to those regions. The advantage of this method lies on its ability to detect the noise value and replace it with a value very close to the original one. The philosophy of median filtering to noise reduction is not restricted to restoration of images alone, and it has also been extended to solve scratch noises and other distortions in speech and audio signals ([1],[2],[3]).

There are various alterations available in literature for reducing the drawbacks of the median filter ([4],[5]). J. S. J. Li and S. Luthi, [6] proposed a multi-shell median filter (MMF) which is genuinely effective. C. J. Juan, [7] suggested a modification to the multi-shell median filter to decrease the processing time and to reduce unnecessary replacements. Nevertheless, under certain specific conditions to be discussed in the following sections, they fail to perform the desired filtering operation. The improved 2-D median filters developed by F. Ghani and E. Khan [8] suffers from the same drawback.

The approach used in this paper represents a substantive departure from the conventional unbiased approach of removing impulse noise. Essentially our contention is that the conventional filtering techniques are unsuited for dealing with impulse noises. Impulses should be removed only by selective identification and suppression [9]. This preserves discontinuities of sufficient duration while eliminating local roughness in the signal.

In noise filtering, the main problem is to preserve some desired signal features while the noise elements are removed. The optimal situation would be when the filter designed is invariant to the desired features and the noise alone is affected.

2 Background

In this paper we have worked with images of size 256x256 and quantized to 256 different gray levels. The filter window size is taken to be an odd integer $(2L+1) \times (2L+1)$ for $L = 0, 1, 2, \ldots$. We define the following signal characteristics [10]:

- An *edge* is a monotonic region between two constant neighborhoods of different value. The connecting monotonic region cannot contain any constant neighborhood.
- An *impulse* is a constant neighborhood followed by at least one, but no more than L points which are then followed by another constant neighborhood. The two boundary points of these L points do not have the same value as the two constant neighborhoods.

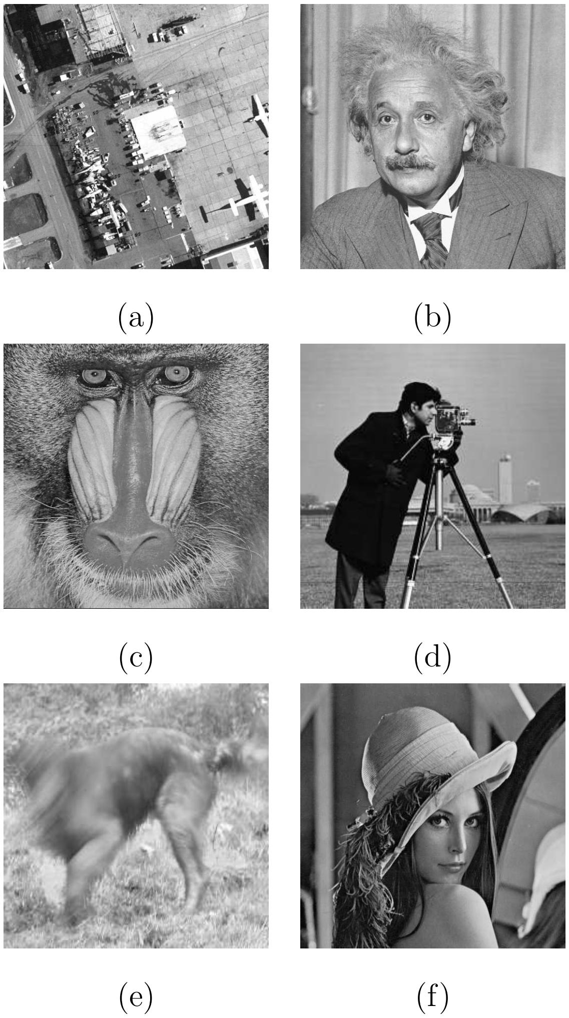


Fig. 1. Original Images used for testing (a) Airfield, (b) Albert, (c) Baboon, (d) Cameraman, (e) Dog (motion artefact), (f) Lena

2.1 Median Filtering (MF):

Median Filtering is a special class of ranked order filters very useful in many signal processing applications. In median filtering, the input pixel is replaced by the median of the pixels contained in a window surrounding it. That is, $v(i,\ j) = median\ \{y(i-k,\ j-l),\ (k,l) \in\ W\}$, where W is a suitably chosen window [11].

The algorithm for median filtering requires arranging the pixel values in the window in increasing or decreasing order and picking the middle value. The window size is generally chosen so that it is odd. The nine-element processing window used for basic 2D-(3x3) median filtering is shown in Table 2.1. A necessary condition for a signal to be invariant under median filtering is that the extended signal should consist of only constant neighborhoods and edges. For a filter length of $(2L+1)$, the nonmonotonic points of the signal within

the window should be separated by at least $(L+1)$ monotonic points. A more rigorous mathematical proofs and detailed analysis of the properties of the median filter is described in [10]. The generalized version of the median filter is the *order statistic filter* described in [12].
Similarly, the efficiency of a 2-D median filter is poor when the number of noise pixels in the window is greater than half the number of pixels in the window [13].

$W_{(-1,-1)}$	$W_{(-1,0)}$	$W_{(-1,1)}$
$W_{(0,-1)}$	$W_{(0,0)}$	$W_{(0,1)}$
$W_{(1,-1)}$	$W_{(1,0)}$	$W_{(1,1)}$

Table 1
A (3x3) processing window W

2.2 Noise Models

Let $a_{i,j}$, for $(i,j) \in \Sigma \equiv \{1,\ldots,M\} \times \{1,\ldots,N\}$, be the gray level of a true $M \times N$ image A at pixel location (i,j), and $[a_{min}, a_{max}]$ be the dynamic range of A, i.e. $a_{min} \leq a_{i,j} \leq a_{max}\ \forall\,(i,j) \in \Sigma$. If a noisy image is denoted by y, for a salt-and-pepper impulse noise model, the observed gray level at pixel location (i,j) is given by

$$y_{i,j} = \begin{cases} a_{min}, & \text{with probability p,} \\ a_{max}, & \text{with probability q,} \\ a_{i,j}, & \text{with probability (1-p-q)} \end{cases} \tag{1}$$

where, $r = p+q$ defines the noise level corrupting the image. The density rate of the noise in an image is designated herein by R which is defined as:

$$R = \frac{MN}{n} \tag{2}$$

MN is the number of total pixels while n is the number of the noisy pixels in the image.

3 Noise Estimation

The first stage in applying a *switching* ([14],[15]) or *decision-based* filter is mapping the noise regions in the corrupted input image. This section describes

the procedure used to estimate and segment the corrupted pixels.
If $E[X] = \widehat{X}$ is the an estimation of the original image obtained after applying the filter to the noisy image Y, the noise candidates are defined as,

$$N_c = \{(i,j) \in A : \widehat{x}_{i,j} \neq y_{i,j} \, and \, y_{i,j} \in \{s_{min}, s_{max}\}\} \tag{3}$$

The set of all uncorrupted pixels $N = A \backslash N_c$.
There are several detectors, which are usually tuned for specific type of edge profiles. Although these detectors are genuinely effective, they cannot be used here because:

- They respond doubly to some noises in the image.
- They cannot distinguish between an edge and an impulse.
- They cannot be used if more than one pixel is corrupted in the same processing window.

In this paper, we use a support vector machine-based detection filter. This takes into account the local features in the vicinity such as the possible presence of details and edges even when the noise is high.

3.1 Local Information based Noise Features

While designing the filter for the noise detection process, there should not be any false detection, and pixels that are genuinely affected should be easily detected. To aid this process, we extract the essential feature vector **O** based on the local information surrounding a pixel.

Definition 1 *Median deviation: The median deviation $\boldsymbol{a}_m$ is a measure of the amount of variation in the input from its median value.*

If $\mathbf{a}_m$ denotes the absolute difference between the input $\mathbf{a}$ and the median value $\mathbf{m}$,

$$\mathbf{a}_m = |\mathbf{a} - \mathbf{m}| \tag{4}$$

$\mathbf{a}_m$ is a measure for detecting the probability whether the input $\mathbf{a}_m$ is contaminated. A large value of $\mathbf{a}_m$ indicates that the input $\mathbf{a}$ is dissimilar to the median value $\mathbf{m}$.

Definition 2 *Successive gradient filter (SGF): The SGF is an enhanced measure of discontinuties and edges in an image.*

The method for using the SGF for isolating the noise was inspired from the similarity between the edge profiles of the impulse noise and that of objects having thin lines. Thin lines correspond to pixels where the brightness function changes abruptly. A change of the image function can be described by a gradient (gradient-based methods) that points in the direction of the largest growth of the image function [16].

Since the characteristics of an impulse error is a sudden fast change, one way to form the detection signal is to observe the difference between successive samples, i.e. the discrete derivative of the signal ([2], [3]). The derivative **d** of the discrete signal **a** can be obtained, according to the definition of the derivative, by dividing the difference of the successive samples by the sampling interval Δt, i.e.

$$d_n = \frac{a_{n+1} - a_n}{\Delta t} \tag{5}$$

Extending the same principle to the two dimensional case, if $a(x, y)$ is a continuous image and $a(i, j)$ be the discrete version of it, then the $(r+1)^{th}$ derivative of $a(x, y)$, $a_y^{r+1}(x, y + \frac{1}{2})$, can be approximated by $a_j^{r+1}(i, j + \frac{1}{2})$ as:

$$a_j^{r+1}(i, j + \frac{1}{2}) = [a_j^r(i, j + 1) - a_j^r(i, j)] \tag{6}$$

To further enhance the detectability, the differentiation can be applied several times. In practice, it has been found that comparison of the central pixel with six of its nearest neighbors is optimal enough to detect the smallest change Table 2. Hence,

$$a_j^6(i, j) = [a_j^5(i, j + \frac{1}{2}) - a_j^5(i, j - \frac{1}{2})] \tag{7}$$

$$\begin{aligned} a_j^6(i, j) = [&a(i, j - 3) - 6a(i, j - 2) + 15a(i, j - 1) - 20a(i, j) + \\ &15a(i, j + 1) - 6a(i, j + 2) + a(i, j + 3)] \end{aligned} \tag{8}$$

The scaling factors associated with the above equations have been omitted. If

$$\begin{aligned} \mathbf{a}_j = [&a(i, j - 3), a(i, j - 2), a(i, j - 1), a(i, j), a(i, j + 1), \\ &a(i, j + 2), a(i, j + 3)]^T, \ \forall i = [1, M], \ j = [4, N - 3] \end{aligned} \tag{9}$$

For impulse noise detection, central difference value is given by:

$$\Delta^6 a_n = a_{n-3} - 6a_{n-2} + 15a_{n-1} - 20a_n + 15a_{n+1} - 6a_{n+2} + a_{n+3} \tag{10}$$

where a_n is the intensity of the pixel under study, and a_i, $i = (n-3)\,to\,(n+3)$ are the intensity of neighboring pixels. The weights from (10) are thus:

$$\mathbf{w} = [1, -6, 15, -20, 15, -6, 1]^T \tag{11}$$

then the output A_H^6 of the gradient filter is given as,

$$a_j^6(i,j) = \mathbf{a}_j \otimes \mathbf{w} \tag{12}$$

where, $\otimes$ denotes 1D convolution.

$$\sum_{j=0}^{L-1} w_j = 0 \tag{13}$$

where, L is the length of the filter.
The noise enhancement can be interpreted in the frequency domain as well. The result of the Fourier transformation is a combination of harmonic functions. The derivative of the harmonic function $Sin(nx)$ is $nCos(nx)$, thus the higher the frequency, the higher the magnitude of its derivatives [16]. The magnitude response of the weights $\mathbf{w}$ is shown in the Fig. 2.

Window Size	PSNR
3	38.37 dB
5	38.30 dB
7	37.32 dB
9	39.15 dB

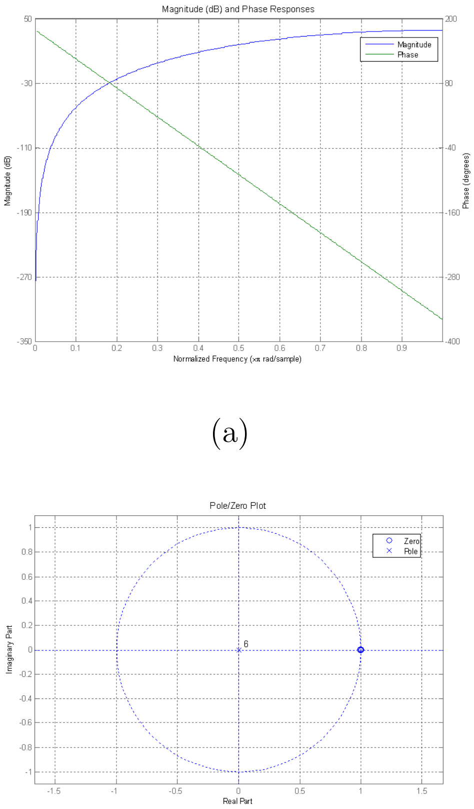
Table 2
Effect of filter window size on PSNR

Definition 3 *Distance Transform: The* distance transform *of an image, measures the relationship between the components of objects that is present in the image.*

Any two-dimensional digital image $a(x, y)$ can be divided into two classes-the object and the background pixels.

$$a(x, y) \in \{Ob, Bg\} \tag{14}$$

The distance transform of an image, $a_d(x, y)$ labels each object pixel of an image with the distance between that pixel and the nearest background pixel.



(a)

(b)

Fig. 2. (a) Magnitude and linear phase response, (b) Pole-zero plot for the filter 11

If $||x, y||$ is a two-dimensional distance metric, then the distance transform of the input image can be defined as in (15).

$$a_d(x,y) = \begin{cases} 0, & \forall a(x,y) \in Bg, \\ \min(||x - x_o, y - y_o||), & \forall a(x_o, y_o) \in Bg, \text{ and } a(x,y) \in Ob \end{cases} .(15)$$

Different distance metrics result in different distance transformations. From our perspective the euclidean distance is the most useful and uses the L_2 norm defined as:

$$||x, y||_{L_2} = \sqrt{x^2 + y^2} \qquad (16)$$

This metric is isotropic and the distances measured are independent of the orientation of the object.
From the above definitions, we can map the input space to the feature space

using the observation vectors, $\mathbf{O}$ as,

$$\mathbf{O} = \{\mathbf{a}_m, \mathbf{a}_H, \mathbf{a}_V, \mathbf{a}_d\} \tag{17}$$

where a_H and a_V denotes outputs from the gradient filter as a result of horizontal and vertical convolution (useful when the information is transmitted as columns instead of rows).

3.2 SVM Classifiers

3.2.1 Support Vector Machines

The Support Vector Machine algorithm [17], operates by mapping the given training set into a possibly high-dimensional feature space and trying to locate in that space a plane that separates the positive from the negative examples. Having found such a plane, the SVM can then predict the classification of an unlabeled example by mapping it into the feature space and calculating the side on which the example lies. The SVM's power comes from its criterion for selecting a separating plane when many candidates planes exist. The SVM chooses the plane that maintains a maximum margin from any point in the training set. Statistical learning theory suggests that, for some classes of well-behaved data, the choice of the maximum margin *hyperplane* will lead to maximal generalization when predicting the classification of previously unseen examples [18].

The advantage of using Support Vector Machines (SVM) for classification problems is that it provides high generalization ability even when the dimension of the input space is very high.

If the training data set contains n examples, each of which is a vector with m features. These vectors can be considered as points in a *m-dimensional* space. In theory, a simple way to build a binary classifier is to construct a *hyperplane* (a plane in a space with more than three dimensions) separating class members (positive examples) from non-members (negative examples) in this space. In general, this hyperplane corresponds to a nonlinear decision boundary in the input space. However, most real-world problems involve non-separable data where the positive examples cannot be distinguished from the negative examples using a *hyperplane*. One solution to this inseparability problem is to map the data into a higher-dimensional space and to define a separating hyperplane there. This higher-dimensional space is called the *feature space*, as opposed to the *input space* occupied by the training examples. With an appropriately chosen *feature space* of sufficient dimensionality the training set can be made separable. Artificially separating the data in this way sometimes exposes the learning system to the risk of finding trivial solutions that over fit the data. This can be avoided by choosing the maximum margin separating

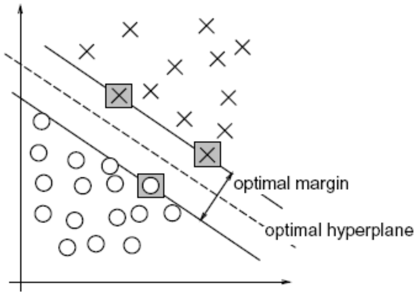


Fig. 3. Linearly separable problem with the optimum separating hyperplane

hyperplane from among the many that can separate the positive from negative examples in the feature space. The decision function for classifying points with respect to the hyperplane involves dot product between the points in the feature space.
For a two-class problems, the optimal separating hyperplane in canonical form must satisfy the following constaints,

$$y_i(wx_i + b) + \eta_i \geq 1 \ \eta_i \geq 0, i = 1, \ldots, l \tag{18}$$

where, $\mathbf{x} \in \Re^n$ is the set of training vectors, and $\mathbf{y} \in \{-1, 1\}$ is the target vector.
The optimal separating hyperplane is determined by the vector w, that minimizes the functional $F(\mathbf{w}, b, \eta)$,

$$\min_{w,b,\eta} C \sum_{i=1}^{l} \eta_i + \frac{1}{2}||w||^2 \tag{19}$$

where C is a suitably chosen parameter. A large C corresponds to assigning a larger penality to errors. Introducing positive *Lagrange* multipliers α_i, to the inequality constraints in model given by (18), we obtain the following dual formulation:

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j \langle x_i x_j \rangle - \sum_{i=1}^{l} \alpha_i \tag{20}$$

$$st \ \sum_{i=1}^{l} y_i \alpha_i = 0, \ 0 \leq \alpha_i \leq C, \ i = 1, \ldots, l \tag{21}$$

The minimum with respect to $\mathbf{w}$, b, η of the functional, F, is given by,

$$\frac{\partial F}{\partial b} = \mathbf{0} \Rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{22}$$

$$\frac{\partial F}{\partial w} = \mathbf{0} \Rightarrow \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i x_i \tag{23}$$

$$\frac{\partial F}{\partial \eta} = \mathbf{0} \Rightarrow \alpha_i + \beta_i = C \tag{24}$$

The solution is then given by $\mathbf{w} = \sum_{i}^{l} \alpha_i y_i x_i$. The free coefficient b can be found from $\alpha_i(y_i(w \cdot x_i + b) - 1) = 0$, $\forall i$, $\alpha_i \neq 0$.

Theorem 1 (Mercer's Theorem) Let K be a continuous, real-valued, positive definite, and symmetric kernel, then (x_i, x_j) in the dual problem (21) can be replaced by a *kernel function* $K(x, y)$. There also exists a mapping, ϕ, such that

$$K(x, y) = \langle \phi(x), \phi(y) \rangle, \; x, y \in \Re_0, \tag{25}$$

The convergence is uniform in x and y. $\Re_0$ is the input space

The use of a kernel function allows the support vector machine to operate efficiently in a nonlinear high-dimensional feature spaces without being adversely affected by the dimensionality of that space. Indeed, it is possible to work with feature spaces of infinite dimension. Moreover, Mercer's theorem makes it possible to learn in the feature space without even knowing the mapping ϕ and the feature space F. The matrix $\kappa_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$ is called the *kernel matrix.*

Kernels are real-valued function $\kappa : \chi \times \chi \rightarrow \Re$, that can be written in the form

$$\kappa(X, Y) = \langle \psi(X), \psi(Y) \rangle \tag{26}$$

for some mapping ψ from χ to a Hilbert space [17].

There are three kernel function usually used in the SVM:

- linear: $x_i^T x_j$,
- polynomial: $(\gamma \langle x_i^T x_j \rangle + r)^p$, $\gamma > 0$, $r \neq 0$
- radial basis function (RBF): $e^{(-\frac{||x_i - x_j||^2}{2\sigma^2})}$
- tangent hyperbolic (sigmoid): $tanh(\beta_0 x_i^T x_j + \beta_1)$

When we use SVM for each selection of *kernel* function, there are some parameters whose values can be altered. These parameters are: trade off cost constant-C, spread-σ (for RBF kernel function), degree of polynomial-p (for polynomial function) or parameters β_0 and β_1 (for sigmoid). The choice of the non-linear mapping function or *kernel* is very important in the performance of the SVM. It was found that, for our application, polynomial kernels, with the C, γ, p and r parameters defined in Table 3, gives the best classification performance in comparison to the sigmoid and the RBF kernels. The value of r avoids problems with the hessian becoming zero. The nonlinear discriminant function is of the form,

Parameters	Value
C	1000
γ	0.001
p	1
r	1

Table 3
Polynomial Kernel Parameters

$$f(\mathbf{x}) = sign(\sum_{i=1}^{l} \alpha_i y_i \kappa(x_i, \mathbf{x}) + b) \tag{27}$$

The final output vector from the discriminant function, is used to map the construct the binary image A_b, where

$$a_b(i,j) = \begin{cases} 1, & \text{if } (i,j) \in N_c \\ 0, & \text{if } (i,j) \in N \end{cases} \tag{28}$$

and $A = \{a(i,j)|(i,j) \in N_c \cup N\}$.
If the input pixel is determined to be a corrupted pixel, it is replaced by its estimated value. The algorithm for this replacement or restoration procedure is discussed in the subsequent sections.

4 Filtering

4.1 Center Weighted Median (CWM) Filter [4]:

The output from a CWM filter, is the median of the window W (Table 2.1) obtained after a weighted adjustment to the central or the original pixel $a(i,j)$. Before understanding the properties of the CWM, it is necessary to define the Weighted Median (WM) Filter.

Definition 4 *Weighted Median (WM) Filter: The weighted median output for any input image A is,*

$$\hat{a}(i,j) = med\{h(k,l) \diamond a(i-k,j-l)|(k,l) \in W_{2L+1}\} \tag{29}$$

$med\{\cdot\}$ denotes the median operation defined in Section 2 and $\diamond$ denotes duplication operator and $h(k,l)$ is defined as,

$$\{h(k,l)|(k,l) \in W_{2L+1}, \sum_{(k,l)\in W_{2L+1}} h(k,l) = c\} \tag{30}$$

$c \geq (2L+1)$ is an odd integer.

In obtaining the output $\hat{a}(i,j)$, the WM filter generates $h(k,l)$ copies of $a(i-k,j-l)$ for each $(k,l) \in W_{2L+1}$, a total of c sample values. Then the median value of the c samples is taken.
The WM filter with central weight $h(0,0) = 2K+1$ and $h(k,l) = 1$ for each $(k,l) \neq (0,0)$ is called the CWM filter. $K > 0$ and is an integer.
The output $\hat{a}(i,j)$ from the CWM filter is given as,

$$\hat{a}(i,j) = med\{a(i-k,j-l), 2K \diamond a(i,j)|(k,l) \in W_{2L+1}\} \tag{31}$$

$$\hat{a}(i,j) = med\{a(i-k,j-l), \underbrace{a(i,j)}_{2Ktimes}|(k,l) \in W_{2L+1}\} \tag{32}$$

when $K = 0$, (32) becomes the median filter (MF), and when $(2K+1) \geq (2L+1)$, then it becomes the identity filter (no filtering). The value of the central weight is a trade-off between the choice of detail preservation and noise suppression. Obviously, the CWM with large central weights performs better in detail preservation but worse in noise suppression in comparison to the one with a smaller central weight. The proofs for the noise suppression and detail preserving capability of the CWM was shown in [4]. Even though there are other modifications to the CWM, like the multi-stage adaptive CWM filter [5], we prefer the CWM for its computational simplicity and equally efficient performance.
It was numerically shown in [19] that the weighted mean filter with a large center weight ($\{\frac{h(0,0)}{h(k,l)} \approx 100|(k,l) \in W_{2L+1}, (k,l) \neq (0,0)\}$), is useful in detecting impulse noise among background image regions with similar frequency levels. Similarly, it was also shown that the successive gradient filter (SGF) (11), as an extension to the weighted average (WA) filter, is useful for salt and pepper noise detection among lower and higher background signal frequency levels respectively.

4.2 Long Range Correlation Filter [20]:

The long range correlation method [20] for image restoration assumes that there exists abundant long-range correlation within natural images, and the human visual system (HVS) can sufficiently utilize this information to process the information.

The best neighborhood matching (BNM) filter implementation of these features of the HVS involves five steps:

(1) *Fetching*: Extract a window W with L^2 pixels from the image which is called a local window. This window may be of any shape, the only restriction is that all the pixels are contiguously connected. For each pixel in this local window, we can get its corresponding flag value $a_{b,W}$. The set of pixels with $a_{b,W} = 1$ is shown as the damaged information in the local window.

(2) *Searching*: Search for another window W' (remote) in the image which is of exactly the same shape and size as the local window W. The flag value of each pixel in window W' is denoted as $A_b^{W'}$. Since the windows W and W' are of the same shape and size, we can find an indexing method that makes every pixel in correspond to a unique pixel at the same position in and vice versa. Sometimes, we restrict the searching procedure to be conducted in a region search range not very far from the local window.

(3) *Matching*: Try to match the remote window W' to the local window W. The matching method is determined by a luminance transformation function v that transforms every pixel in W'.

The pixel pairs in the two windows can be classified into three categories. In the first category, the corresponding pixels in W' and W are good ($a_{b,W'} = 0$ and $a_{b,W} = 0$). In the second category, the pixel in W' is good, but its corresponding pixel in W is bad ($a_{b,W'} = 0$ and $a_{b,W} = 1$). In the last category, the pixel in W is bad irrespective of whether its corresponding pixel in W is good or not ($a_{b,W'} = 1$, $a_{b,W} = 0$ or $a_{b,W} = 1$). The pixels in the first category compose the matching part of the window. The number of pixels in this category is

$$n_M = \sum_{j=}^{L} \sum_{i=1}^{L} [1 - a_{b,W}(i,j)][1 - a_{b,W'}(i,j)] \tag{33}$$

The matching result is evaluated by the mean-squared error of the matching part between the transformed window and the window.

$$MSE_M = \frac{1}{n_M} \sum_{i=1}^{L} \sum_{j=1}^{L} [1 - a_{b,W}(i,j)][1 - a_{b,W'}(i,j)][l_{i,j} - v(W'(i,j))]^2 \tag{34}$$

The MSE_M (34) is to be as small as possible.

(4) *Competing*: Each candidate remote window in the search range will result in its corresponding MSE_M. An obvious effective standard for the selection of remote windows is to choose the one with the least MSE_M.

(5) *Recovering*: Recover the damaged pixels in the local window using the good pixels in the transformed remote window. Suppose we have the winning remote window and its related matching transformation function. Because the remote window and the local window are very well matched, some bad pixels in W can be recovered by using their corresponding good pixels in W'. The renewed local window is copied back into its corresponding position in the recovered image and modify the flag values of the recovered pixels from 1 to 0.

By applying the five steps above, some bad pixels ($a_{b,W} = 1$ and $a_{b,W'} = 0$) in the contaminated image are recovered. These steps have to be followed recursively for restoration of the image until the flags of all the pixels in the image are all 0.

4.3 Modified Multi-shell median filter (MMMF) [7]:

Definition 5 *The output from the MMMF filter*

$$\hat{a}(i,j) = \begin{cases} Max(a_{-1,0}, a_{1,0}), & \textit{if } a_{0,0} - \max[S] \geq \beta \\ Min(a_{-1,0}, a_{1,0}), & \textit{if } \min[S] - a_{0,0} > \beta \\ a_5, & \textit{otherwise} \end{cases} \tag{35}$$

where, $\beta = 16$ is a fixed threshold ([21]), and
$S = \{a_{k,l} | -1 \leq k \leq 1, -1 \leq l \leq 1; (k,l) \neq (0,0)\}$.

The MMMF [7] suffers from a serious limitation, for regions in the image corresponding to low frequency signal and corrupted with a similar type of noise (negative impulse), (35) is not satisfied. Under such conditions, this sort of noise is not eliminated as shown in Fig. 4(a). The switching filter on the other hand does not suffer from this drawback and can eliminate this type of noise successfully as shown in Fig. 4(b).

While the implementation of both CWM and MMMF is computationally less intensive, it is important to realize that the performance of the CWM deteriorates rapidly as the noise density increases beyond 30% (Fig. 9). Thus, the switching is preferred between the BNM and the MMMF rather than a combination of CWM and MMMF.

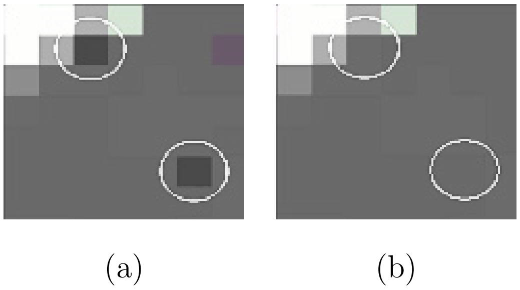


Fig. 4. (a) White circle shows the noise left behind by a 3*3 MMMF [7], (b) The circles now indicate the position where the noise was removed after using the proposed scheme.

5 Numerical Results

5.1 Training the SVM classifier

The optimal separating hyperplane for distinction between the two classes is obtained by using a set of supervised class labels for each of the training images (corrupted by 40% impulse noise) in Fig. 5. The inputs to the training model will be the set of unsupervised features O extracted by the procedure discussed in section 3. The algorithm for isolation of the noise was tested and

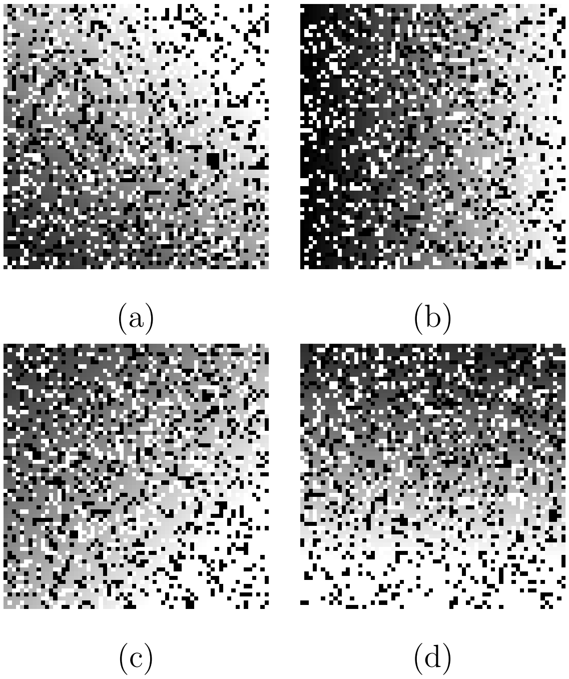


Fig. 5. Example training images for 40% noise

Fig. 6 shows an enlarged view of the image before filtering, after isolation of noise, and finally after removing the noise. The final output image, after filtering the cameraman image (Fig. 1(d)) corrupted with 4% noise density is given in Fig. 7(d). The noise segmentation statistics for the *airfield* (worst

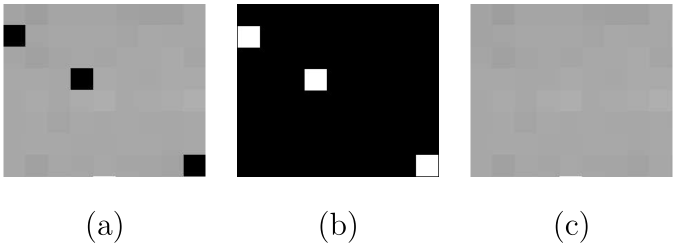


Fig. 6. (a) The original image corrupted with impulse noise, (b) After isolating the noise, (c) After filtering the noisy image.

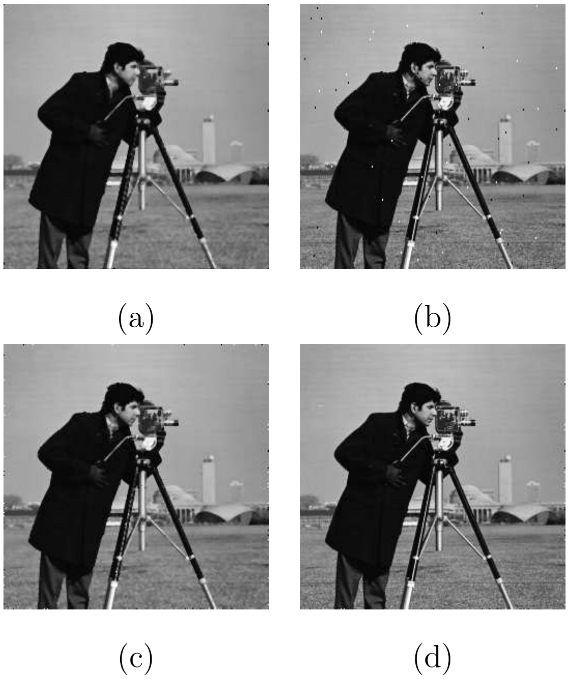


Fig. 7. Restoration of the *Cameraman* (256x256 scanned at 72 dpi & 8 bits/pixel) corrupted with with 4% noise (a) After MF (SSIM=0.8711, PSNR=13.55dB), (b) After MMMF (SSIM=0.9117, PSNR=29.12dB), (c) After CWM (SSIM=0.92744, PSNR=29.53dB), (d) After filtering using our method (SSIM=0.9931, PSNR=37.32dB)

case scenario) test image are:

- Accuracy: 0.9827
- Precision: 0.9665
- Specificity: 0.9653
- Sensitivity: 1.0000
- F-score: 0.9830

It was observed that among the images shown in Fig. 5, the training image Fig. 5(b) gives the best classification rate.

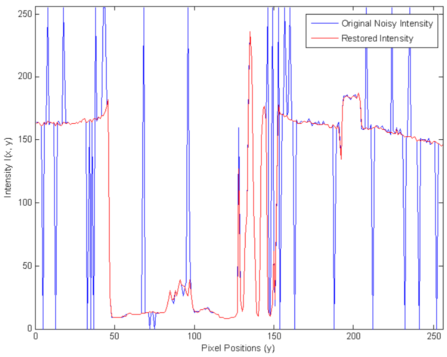


Fig. 8. Intensity slice plot for the cameraman image before and after restoration

Actual\Predicted	Noise	Non-noise
Noisy pixels	0.8663	0.1337
Non-noisy pixels	0.1214	0.8786

Table 4
Confusion Matrix for noise recognition in *cameraman* using the method proposed in [21]

Actual\Predicted	Noise	Non-noise
Noisy pixels	1.0000	0
Non-noisy pixels	0.0000	1.0000

Table 5
Confusion Matrix for noise recognition (in *albert*, *baboon*, *cameraman*, *dog*, *lena*) using our method

5.2 Evidence of functional improvement:

The performance improvement of the propounded method over the existing one is presented here. The evaluation of the performance can be based on '*Subjective Fidelity Criterion*', PSNR (36), or the structural similarity based quality measure (SSIM index [22]).

The suggested scheme is found to remove noises even from the edges of an image and preserve the detail regions Fig. 4. The Fig. 4(a), shows a portion of the *Cameraman* image that was affected by impulse noise, and after administering the MMMF. The white circles highlight is used for highlighting the negative impulses left behind after MMMF filtering. While Fig. 4(b) shows the same region after using our method. It should be known that it may not be possible to remove some noise even by repeated filtering if the root signal also has added noise [10]. Fig. 9 shows the different Peak Signal to Noise Ratio (PSNR) for different noise densities, calculated using (36) and Table 6 compares the PSNRs for the different test images.

$$\text{PSNR} = 20 \left[\log_{10} \left[\frac{MN\ max(\hat{A})}{\sqrt{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [\hat{a}(i,j) - a(i,j)]^2}} \right] \right] \tag{36}$$

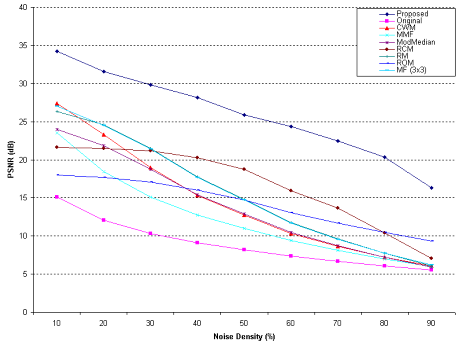


Fig. 9. The improvement in PSNR in the proposed approach as against the original noisy image.

Natural image signals are highly structured: their pixels exhibit strong dependencies, especially when they are spatially proximate, and these dependencies carry important information about the structure of the objects in the visual scene. Most error metrics are based on pointwise signal differences, which are independent of the underlying signal structure. There are some others that are based on error sensitivity decompose image signals using linear transformations, these do not remove the strong dependencies. However, the method referenced below is a more direct way to compare the structures of the reference and the distorted signals. This will give us a measure of the level of structural information preservation (or perceivable distortion) after filtering. The SSIM index that was described above, is a function of three factors: the *luminance*, *contrast* and *structural* similarity. Thus,

$$SSIM(A, \hat{A}) = f(l(A, \hat{A}), c(A, \hat{A}), s(A, \hat{A})) \tag{37}$$

The SSIM index in specific form can be written down as (38):

$$SSIM(A, \hat{A}) = \frac{(2\mu_A \mu_{\hat{A}} + C_1)(2\sigma_{A\hat{A}} + C_2)}{(\mu_A^2 + \mu_{\hat{A}}^2 + C_1)(\sigma_A^2 \sigma_{\hat{A}}^2 + C_2)} \tag{38}$$

where, $C_1 = (K_1 L)^2$, $C_2 = (K_2 L)^2$, L is the dynamic range of the pixel intensities (255 for 8-bit grayscale images), $K_1 \ll 1$ and $K_2 \ll 1$ is a small constant. For our case, we use the following parameter settings: $K_1 = 0.01$; $K_2 = 0.03$.

In practice, one only requires a single overall integrity measure for the entire image. The Mean SSIM (MSSIM) index is thus used as a measure to evaluate the overall consistency:

$$MSSIM(A, \hat{A}) = \frac{1}{M} \sum_{j=1}^{M} SSIM(\mathbf{a}_j, \hat{\mathbf{a}}_j) \tag{39}$$

where A, $\hat{A}$ are the original and restored images, respectively; $\mathbf{a}_j$ and $\hat{\mathbf{a}}_j$ are the image contents at the j^{th} local window; and M is the total number of local windows of the image.

Filtering Algorithm	**Airfield**	**Albert**	**Baboon**	**Cameraman**	**Dog**	**Lena**
Original	9.134	9.555	9.577	9.111	9.427	9.183
MF (3×3)	17.90	18.93	17.41	17.67	18.91	18.75
MF (5×5)	22.08	26.11	20.30	21.92	26.30	26.67
ROM (3×3) [23]	13.21	14.81	16.1	15.98	13.55	17.87
RM $(L = 3, U = 9)$ [24]	18.07	19.10	17.50	17.89	19.07	18.94
RCRS $(L = 4, U = 6)$ [25]	17.97	24.54	19.61	20.28	23.32	24.55
MMMF (3×3) [6]	12.77	13.27	13.03	12.73	13.25	12.95
Proposed	23.40	30.22	24.68	28.15	30.54	33.17

Table 6
Comparative Restoration results in PSNR (in dB) for the test images corrupted with 40% impulse noise.

Filtering Algorithm	**Airfield**	**Albert**	**Baboon**	**Cameraman**	**Dog**	**Lena**
Original	0.080	0.041	0.095	0.067	0.037	0.041
MF (3×3)	0.4451	0.370	0.400	0.440	0.454	0.458
MF (5×5)	0.580	0.571	0.441	0.729	0.697	0.766
ROM (3×3) [23]	0.318	0.276	0.270	0.389	0.258	0.398
RM $(L = 3, U = 9)$ [24]	0.455	0.379	0.406	0.460	0.466	0.470
RCRS $(L = 4, U = 6)$ [25]	0.461	0.508	0.327	0.654	0.574	0.690
MMMF (3×3) [6]	0.203	0.122	0.207	0.163	0.143	0.137
Proposed	0.824	0.829	0.831	0.912	0.908	0.920

Table 7
Comparative Restoration results in SSIM Index [22] for the test images corrupted with 40% impulse noise.

6 Conclusions

A new SVM based algorithm for noise detection is presented in this paper and a multi-shell median approach to filtering the data was developed. The effectiveness of the proposed approach relies on the filter capability to detect the true noise configurations. The real power of the noise segmentation can be realized from the fact that the noise detection rate was 100% for (0 − 90%) noise density for the *cameraman* test image (training images corrupted with 40% noise). The developed filter is adaptive and preserves edges in the image. The optimal goal of the filter modeled is to improve the quality of the image by noise reduction. However, as the complexity of the system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost exclusive characteristics.

References

[1] T. Kasparis, J. Lane, Adaptive scratch noise filtering, Consumer Electronics, IEEE Transactions on 39 (4) (1993) 917–922.

[2] C. Chandra, M. S. Moore, S. K. Mitra, An efficient method for the removal of impulse noise from speech and audio signals, in: Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on, Vol. 4, 1998, pp. 206–208.

[3] I. Kauppinen, Methods for detecting impulsive noise in speech and audio signals, in: Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on, Vol. 2, 2002, pp. 967– 970.

[4] S. J. Ko, Y. H. Lee, Center weighted median filters and their applications to image enhancement, Circuits and Systems, IEEE Transactions on 38 (9) (1991) 984–993, 0098-4094.

[5] T. Sun, M. Gabbouj, Y. A. Neuvo, Image restoration by multistage adaptive center weighted median filters, in: Proc. SPIE Vol. 2180, p. 99-110, Nonlinear Image Processing V, Edward R. Dougherty; Jaakko T. Astola; Harold G. Longbotham; Eds., 1994, pp. 99–110.

[6] J. S. J. Li, S. Luthi, A real-time 2-D median based filter for video signals, Consumer Electronics, IEEE Transactions on 39 (2) (1993) 115 – 121.

[7] J. Chang Jung, Modified 2-D median filter for impulse noise suppression in a real-time system, Consumer Electronics, IEEE Transactions on 41 (1) (1995) 73 – 80.

[8] F. Ghani, E. Khan, Missing lines recovery and impulse noise suppression using improved 2-D median filters, Consumer Electronics, IEEE Transactions on 45 (2) (1999) 356 – 360.

[9] R. Bernstein, Adaptive nonlinear filters for simultaneous removal of different kinds of noise in images, Circuits and Systems, IEEE Transactions on 34 (11) (1987) 1275.

[10] N. C. Gallagher, G. L. Wise, A theoretical analysis of the properties of median filters, IEEE Trans. Acoust., Speech, Signal Processing ASSP-29 (1981) 1136–1141.

[11] A. K. Jain, Fundamentals of Digital Image Processing, Prentice Hall, Englewood Cliffs, NJ, USA, 1989.

[12] T. S. H. Alan C. Bovik, D. C. Munson, A generalization of median filtering using linear combinations of order statistics, IEEE Trans. ASSP ASSP-31 (1983) 1342–1350.

[13] R. C. Gonzalez, R. E. Woods, Digital image Processing, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[14] E. How-Lung, M. Kai-Kuang, Noise adaptive soft-switching median filter, Image Processing, IEEE Transactions on 10 (2) (2001) 242–251, 1057-7149.

[15] C. Tao, W. Hong Ren, Space variant median filters for the restoration of impulse noise corrupted images, Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on] 48 (8) (2001) 784–789.

[16] M. Sonka, V. Hlavac, R. Boyle, Image Processing: Analysis and Machine Vision, O'Reilly, 1999.

[17] B. Schölkopf, A. J. Smola, Learning with Kernels: Support Vector Machines, Regularization , Optimization, and Beyond, MIT Press, Cambridge, MA, 2002.

[18] V. Vapnik, The Nature of Statistical Learning Theory, Sptinger-Verlag, New York, 2000.

[19] P. Pankajakshan, V. Kumar, On adaptive noise segmentation and filtering, in: Control, Instrumentation and Information Communication (CIIC01). Proc. International Conf. on,, Kolkata, India, 2001.

[20] D. Zhang, W. Zhou, Image information restoration based on long-range correlation, in: Circuits and Systems for Video Technology, IEEE Transactions on, Vol. 12, 2002, pp. 331–341, 1051-8215.

[21] T.-C. Lin, P.-T. Yu, Adaptive Two-Pass Median Filter Based on Support Vector Machines for Image Restoration, Neural Comp. 16 (2) (2004) 333–354.

[22] W. Zhou, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, Image Processing, IEEE Transactions on 13 (4) (2004) 600– 612, 1057-7149.

[23] K. M. Singh, P. K. Bora, S. B. Singh, Rank-ordered mean filter for removal of impulse noise from images, in: Industrial Technology, 2002. IEEE ICIT '02. 2002 IEEE International Conference on, Vol. 2, 2002, pp. 980– 985.

[24] A. B. Hamza, P. L. Luque-Escamilla, J. Martnez-Aroza, R. Romn-Roldn, Removing noise and preserving details with relaxed median filters, Journal of Mathematical Imaging and Vision 11 (2) (1999) 161 – 177.

[25] R. C. Hardie, K. E. Barner, Rank conditioned rank selection filters for signal restoration, Image Processing, IEEE Transactions on 3 (2) (1994) 192–206, 1057-7149.