A two-stage shearlet-based approach for the removal of random-valued impulse noise in images

Guorong Gao[a], Yanping Liu[a], Demetrio Labate[b,*]

[a] *College of Science, Northwest A&F University, Yangling, Shaanxi, 712100 China*
[b] *Department of Mathematics, University of Houston, Houston, TX 77204, USA*

Abstract

In this paper, we introduce a novel two-stage denoising method for the removal of random-valued impulse noise (RVIN) in images. The first stage of our algorithm applies an impulse-noise detection routine that is a refinement of the HEIND algorithm and is very accurate in identifying the location of the noisy pixels. The second stage is an image inpainting routine that is designed to restore the missing information at those pixels that have been identified during the first stage. One of the novelties of our approach is that our inpainting routine takes advantage of the shearlet representation to efficiently recover the geometry of the original image. This method is particularly effective to eliminate jagged edges and other visual artifacts that frequently affect many RVIN denoising algorithms, especially at higher noise levels. We present extensive numerical demonstrations to show that our approach is very effective to remove random-valued impulse noise without any significant loss of fine-scale detail. Our algorithm compares very favourably against state-of-the-art methods in terms of both visual quality and quantitative measurements.

Keywords: Image denoising; inpainting; morphological component analysis; random-valued impulse noise; shearlets; wavelets.

1. Introduction

Random-valued impulse noise (RVIN) is a common cause of image degradation, frequently found in images acquired from digital cameras and is usually due to malfunctioning camera sensors, faulty memory locations in hardware, or transmission in a noisy channel. RVIN is characterized by the alteration of specific pixels in an image with the result that their intensity values become incompatible with the neighbouring pixels [1]. The presence of impulse noise may severely corrupt the information embedded in the original data and it is

[*] Corresponding author (Tel.: +1 713-7433492). Guorong Gao and Yanping Liu are the joint first authors.

Email addresses: lypggr@sina.com (Guorong Gao), ypliu0626@sina.com (Yanping Liu), dlabate@math.uh.edu (Demetrio Labate) critical to correct the image degradation before subsequent image processing tasks such as edge detection, feature extraction or classification.

Non-linear filters such as median (MED) filters are popular techniques for removing RVIN because of their simplicity and low computational cost [1, 2]. However, conventional median filters apply the median operation unconditionally, that is, without discriminating between corrupted and uncorrupted pixels. As a result they modify both noisy and noise-free pixels alike causing loss of image detail that may be especially significant at higher noise levels.

To overcome these limitations, an effective strategy consists in applying first an impulse-noise detection routine so that only the noisy pixels would undergo a filtering process. Several *two-stage* denoising algorithms have been proposed based on this processing strategy, most notably the directional weighted median filer (DWM) [3], the switching median filter with boundary discriminative noise detection (SM-BDND) [4], the direction-based adaptive weighted switching median filter (DAWSM) [1] and the homogeneous amount based (HAB) filter [5]. The efficiency of these two-stage denoising schemes clearly depends on the combined efficiency of impulse detection and filtering routines used. The DWM, in particular, uses a direction-based approach to detect noisy pixels that computes the difference between a pixel and its neighbors within a window of size 5; the detected noisy pixels are then replaced by the output of a directional weighted median filter [1]. The SM-BDND algorithm applies the boundary discriminative noise detection method to detect noisy pixel positions; detected noisy pixels are then replaced by the median value of the pixels in the filtering window. The DAWSM algorithm uses a sophisticated impulse-noise detection method, called Highly Effective Impulse Noise Detection algorithm (HEIND), originally proposed by Duan et al. [6]. HEIND uses both boundary and directional information to detect noisy pixels; after the detection stage, the noisy pixels are replaced by the weighted median value of uncorrupted pixels in a directional filtering window. The Fuzzy Weighted Non-Local Means method (FWNLM) proposed by Wu et al. [7] that uses a fuzzy weighting function with the non-local means algorithm to selectively pick pixels when calculating the pixel similarity. We also recall the multistage denoising method by two of the authors in [8], which uses a simple directional filter similar to the method in [2] to detect noisy pixels, followed by a total-variation inpainting routine[1].

Although multistage RVIN denoising schemes perform consistently better than conventional median filters, they frequently produce jagged edges and introduce visual artifacts, such as block effects, in the processed images especially at higher noise levels. This limitation is due to the fact that, even though these algorithms can efficiently estimate the location of noisy pixels and enforce compatibility among neighboring pixels, they are usually not very efficient in recovering the *geometry of the corrupted data*. This is particularly evident at

[1] This algorithm is specifically designed to handle salt and pepper noise, and both its noise-detection and inpainting routines are much less sophisticated than those that will be considered in this paper.

higher noise levels, where the problem of recovering the information at the locations of corrupted pixels is more challenging. This observation is indeed the motivation for some recent work in [9], where the denoising algorithm includes a routine designed to recover the edge information, and [5], where the task to restore the corrupted information is handled by a total-variation inpainting method.

In this paper, we propose a novel two-stage algorithm for restoring data affected by RVIN which includes, as a first stage, a very efficient impulse-noise detector that we obtain as a refinement of the HEIND algorithm and, as a second stage, an innovative shearlet-based image inpainting routine to restore the corrupted pixels. The main advantages of our approach are due to (1) the improved method for impulse-noise detection and (2) the application of the image inpainting routine, taking advantage of the properties of shearlets to efficiently recover the geometry of the original image. The shearlet representation, introduced by one of the authors and his collaborators, is an advanced multiscale method providing optimal approximation properties for images with edges. It was recently proven that shearlets have a superior ability to recover occluded edges and their application to image inpainting can significantly outperform other conventional inpainting algorithms [10].

As a part of this work, we numerically analyze the performance of our denoising algorithm and present extensive numerical demonstrations on images degraded by two classical types of random-valued impulse noise with a wide range of noise levels varied from 10% to 80%. We compare our algorithm against a range of conventional and state-of-the-art denoising schemes, including the median filer (MED), the modified noise adaptive soft switching median filter (MNASM) with HEIND detector [6] (H-MNASM), the SM-BDND method[4], the FWNLM method [7] and the DAWSM method [1]. We have also included the comparison with a version of our algorithm containing another more traditional inpainting method, based on total variation. The experimental results reported in this paper show that our algorithm is extremely competitive for RVIN denoising and outperforms current state-of-the-art algorithms in terms of visual quality and objective measures. In particular, our approach virtually eliminates jagged edges and other visual artifacts which are frequent in other RVIN denoising algorithms, especially at higher noise levels.

The rest of the paper is organized as follows. In Section 2, we present our new two-stage denoising algorithm, combining an improved method for the detection of RVIN (Section 2.2) and a powerful image inpainting method (Section 2.3). In Section 3, we provide extensive numerical experiments to validate our algorithm even under very challenging noise conditions and compare our results against several other state-of-the-art denoising methods. We conclude with some remarks in Section 4.

2. Proposed random-valued impulse noise detection method

As indicated above and illustrated in Figure 1, in this paper we adopt a two-stage denoising strategy. The first stage of our algorithm is devoted to detect the location of the pixels affected by impulse noise. Once the noisy pixels are detected, they are handled as missing pixels that need to be restored. Hence, the second stage of our algorithm is an image inpainting routine designed to recover the information at the noisy pixel locations.

Before presenting the detailed description of the two stages of our algorithm, we briefly review the noise models that are considered in this paper.

2.1. Impulse-noise models

Four impulse-noise models are usually described in the literature [1, 4].

1. Noise Model 1. The simplest impulse-noise model is the *salt-and-pepper* noise where pixels are randomly corrupted by two fixed extreme values, usually 0 and 255 (for 8-bit grayscale images), with the same probability. That is, for each pixel location (i, j) in the image, with intensity value $s_{i,j}$, the corresponding pixel in the noisy image is $x_{i,j}$ where the probability density function of $x_{i,j}$ is

$$f(x) = \begin{cases} p/2 & \text{for } x = 0 \\ 1 - p & \text{for } x = s_{i,j} \\ p/2 & \text{for } x = 255 \end{cases}$$

   and p is the noise density $(0 < p < 1)$.

2. Noise Model 2. It is similar to Model 1, with the difference that the probabilities of "salt" and "pepper" are unequal. That is, the probability density function of $x_{i,j}$ is

$$f(x) = \begin{cases} p_1 & \text{for } x = 0 \\ 1 - p & \text{for } x = s_{i,j} \\ p_2 & \text{for } x = 255 \end{cases}$$

   where $p = p_1 + p_2$ with $p_1 \neq p_2$ (and $0 < p < 1$).

3. Noise Model 3. Instead of taking two *fixed values*, impulse-noise is modeled by two *fixed ranges* of same length m, that appear at both ends of the integer range. This definition provides a more realistic modeling of noise found in practical applications, such as medical imaging [4, 11]. In this setting, the probability density function of $x_{i,j}$ is

$$f(x) = \begin{cases} p/2 & \text{for } 0 \leq x < m \\ 1 - p & \text{for } x = s_{i,j} \\ p/2 & \text{for } (255 - m) < x \leq 255 \end{cases}$$

   where p is the noise density $(0 < p < 1)$.

4. Noise Model 4. It is similar to Model 3, with the difference that the probabilities of low intensity impulse noise and high intensity impulse noise are unequal. Hence the probability density function of $x_{i,j}$ is

$$f(x) = \begin{cases} p_1 & \text{for } 0 \leq x < m \\ 1-p & \text{for } x = s_{i,j} \\ p_2 & \text{for } (255-m) < x \leq 255 \end{cases}$$

   where $p = p_1 + p_2$ with $p_1 \neq p_2$ (and $0 < p < 1$).

Note that Models 1 and 2 are special cases of Models 3 and 4, corresponding to $m = 1$. Since they are more general models, in the rest of the paper we will consider Noise Models 3 and 4 for the discussion of our algorithms.

2.2. First Stage: Impulse-Noise Detection

The first stage of our algorithm is an impulse noise detector that we derive as a refinement of the Highly Effective Impulse Noise Detection (HEIND) originally introduced by Fei Duan et al. [6]. In our approach, which we call Improved Impulse Noise Detection (IIND), we generates a first estimate of the locations of the noisy pixels using the HEIND routine; based on this detection result, we calculate an estimate of the denoised image using a simple filtering procedure; next we use this estimate of the denoised image as a new input of the HEIND routine and repeat the same procedure again. In the following, we describe the main ideas of the HEIND algorithm and our modifications, including a discussion of the performance of this procedure.

2.2.1. The HEIND algorithm

The HEIND algorithm uses a coarse-to-fine scale strategy that consists in examining, for each pixel, the neighbouring pixels over two windows of different sizes. During the coarse-scale analysis, each pixels is examined over a window of size 21×21; this analysis is similar to the detection method in BDND [4]. The examination at the fine-scale level is performed using a window of size 7×7 and this stage uses four directional convolution-type filters. Only those pixels that are classified as ”corrupted” during the coarse-scale analysis are passed to the fine-scale stage filter for examination and only those pixels that are classified as ”corrupted” in both stages are considered be noisy pixels. The examination of the pixels over windows of different sizes has the objective to avoid that noise-free pixels could be misclassified as corrupted ones based on a single-scale examination alone. We refer to [1, 4] for additional detail about the HEIND algorithm.

2.2.2. Improved RVIN detection

Through extensive numerical experimentation, we have found that, after the HEIND detection stage, there are still pixels corrupted by impulse noise that have not been detected, especially for high noise levels. To address this issue, in this paper, we introduce the following iterative procedure. After detecting the noisy pixels using the HEIND routine, we estimate the true value of the noisy pixels using a two-window-based median filter that is described below; next, we use this estimate of the denoised image as a new input of the HEIND routine and repeat the same procedure again. In fact, it turns out that it is sufficient to repeat this iterative procedure just two or three times (we have used 3 iterations for all our tests) to obtain a highly improved detection of the noisy pixels. Finally, the decision function at the pixel location (i,j) is given by:

$$M(i,j) = \begin{cases} 1 & \text{if } u_0(i,j) = u(i,j) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where u is the image intensity and u_0 is the estimate of the noise-free image in the last iteration. The value $M(i,j) = 0$ indicates that the (i,j) pixel is corrupted.

2.2.3. Pixel intensity estimation using a two-window-based median filter

As indicated above, our IIND algorithm includes a routine to estimate the pixel intensity at the locations detected by the HEIND algorithm. This routine applies an adaptive two-window-based median filter based on the method proposed by Hsieh et al. [12]. Hsieh observed that, for low levels of noise, it is sufficient to select a "small" filtering window since noise-free pixels can be easily found among neighbouring points; on the other hand, a "large" window should be employed at high noise levels because neighbouring points are more likely to be also corrupted. Moreover, if noisy points are sparse in the image, it is sufficient to examine the neighboring points along horizontal and vertical directions; on the other hand, for high noise level it is preferable to also examine the neighboring points along the diagonal orientations. As a result, in this work, similar to [12], we use two types of windows, that we call *Type 1* and *Type 2* and are shown in Fig. 2. We use the Type 1 window if the noise density is less than 50%, otherwise we use the Type 2 window. That is, given a noisy pixel location (i,j), the Type 1 and Type 2 windows, denoted by W_1 and W_2, respectively, contain the pixels

$$\begin{aligned} W_1 &= \{u(i,j_1) | \, j_1 = j \pm m\} \cup \{u(i_1,j) | \, i_1 = i \pm m\} \\ W_2 &= \{u(i_1,j_1) | \, i_1 = i \pm m, j - m \leq j_1 \leq j + m\} \\ &\quad \cup \{u(i_1,j_1) | \, i - m \leq i_1 \leq i + m, j_1 = j \pm m\} \end{aligned}$$

The filtering window is then iteratively dilated as long as the number of noise-free pixels in the window, denoted by N_w, is no larger than 1. When the number of noise-free pixels in the window is larger than 1, then the noisy pixel $u(i,j)$ is replaced by the median value of these noise-free pixels in the window. Thus, our adaptive two-window-based median filter algorithm can be summarized as follows: (1.) Evaluate the noise density p and initialize the window size parameter $m = 1$. (2.) If $p < 50\%$ then, for each noisy pixel location (i,j), do the following steps: (i) build a Type-1 window W_1; (ii) if W_1 does not include noise-free pixels then set $m = m + 1$ and go to step (2.)(i); else stop. (3.) If

$p \geq 50\%$ then, for each noisy pixel location (i, j), do the following steps: (i) build a Type-2 window W_2; (ii) if W_2 does not include noise-free pixels then set $m = m + 1$ and go to step (3.)(i); else stop. (4.) Replace the noisy pixel $u(i, j)$ with the median value of these pixels in W_1 or W_2.

Table 1 reports the difference between the number of *true* noisy pixels and the number of *detected* noisy pixels in different noisy images as they are computed using the standard HEIND algorithm and our IIND method. The minus sign means that the number of *detected* noisy pixels is larger than that of the *true* noisy pixels, that is to say, in this case, there exists a small number of noise-free pixels are detected as noisy pixels. The results in Table 1 show that a large number of noisy pixels are misclassified by the HEIND method but are correctly detected by our method. In fact, our method is remarkably effective in identifying corrupted pixels and outperforms very significantly the HEIND method. Notice that IIND shows some false detections for Model 4 noise, this may because there exist a small number of impulse noise lie between [0,m) and (255-m,255] in the original noise free image. However, a small number of false detections is usually not harmful since these pixels are not significantly affected during the filtering process.

2.3. Second stage: restoration of noisy pixels by image inpainting

As mentioned above, one drawback of many impulse-noise denoising schemes is that, even when they are able to accurately detect the noisy pixel locations, they are not very efficient to recover the underlying geometry of the data at the location of the corrupted pixels. Our method to address this task is based on the application of an inpainting strategy that is designed to be, in a precise sense clarified below, as efficient as possible for the recovery of edges and other fine-scale structural features possibly corrupted by noise.

Recall that variational methods have been successfully employed for inpainting [13, 14]. This approach is motivated by the intuitive idea that one can fill the missing 'holes' in an image by propagating information from the boundaries of the holes while guaranteeing smoothness of some sort. The variational approach has been shown to perform well on piecewise smooth images. However, real images also contain textured regions and other fine-scale structures that are not recovered efficiently using variational methods.

One powerful strategy to overcome this limitation is based on the principles of *morphological component analysis* (MCA), whose central idea is to use multiple dictionaries to break up an image into its elementary geometric constituents. Suppose that we want to recover an image x, of size N^2 from it's noisy observation y, and assume that x is sparse in an overcomplete dictionary $\mathcal{D}$, where $\mathcal{D} \in \mathbb{R}^{N^2 \times K}$ is the matrix form of the overcomplete dictionary. That is, writing

$$x = \mathcal{D}\alpha = \sum_{k=1}^{K} \alpha_k \, d_k,$$

where $\mathcal{D} = [d_i, \ldots, d_k]$, $d_k \in \mathbb{R}^{N^2}$, we expect that "many" of the representations coefficients α_k are negligible. If we want to minimize the number of non-negligible coefficients, we can set up the minimization problem [15, 16]

$$\hat{\alpha} = \min \|\alpha\|_1 \quad \text{subject to } \|y - \mathcal{D}\alpha\|_2 \leq \sigma, \tag{2}$$

and obtain $\hat{x} = \mathcal{D}\hat{\alpha}$. Note that, for an appropriate Lagrange multiplier λ, the solution of (2) is exactly the solution of the unconstrained optimization problem[2]

$$\min_{\alpha} \lambda \|\alpha\|_1 + \frac{1}{2} \|y - \mathcal{D}\alpha\|_2^2. \tag{3}$$

We are interested in the situation where the image is a superposition of several components, each one having a sparse representation with respect to a certain dictionary. Hence, we model the image x as $x = \sum_{k=1}^{K} x_k$ and use a dictionary built by amalgamating several subdictionaries $\mathcal{D}_1, \ldots, \mathcal{D}_K$ that are 'incoherent'. That is, each x_k has a sparse representation in the subdictionary $\mathcal{D}_k$ but its representation in the subdictionaries $\mathcal{D}_l$, $l \neq k$, is not sparse. In particular, we can assume that x is a superposition

$$x = x_p + x_t, \tag{4}$$

where x_p and x_t are the piecewise smooth component and textured component of the data, respectively. In this setting, for the subdictionary associated with texture component of the data we can choose a local discrete cosine dictionary, which is sparse for locally periodic patterns. For the piecewise smooth component of the data, we can choose a shearlet or a curvelet dictionary, which are known to be sparse for this type of data. The incoherence of the two dictionaries has been verified heuristically in [17] (using DCT and curvelet dictionaries) and more recently and rigorously in [18]. Then, to achieve the 'geometric separation' (4), we set up the minimization problem:

$$\hat{\alpha}_t, \hat{\alpha}_p = \min_{\alpha_t, \alpha_p} \lambda \left(\|\alpha_t\|_1 + \|\alpha_p\|_1 \right) + \frac{1}{2} \|y - \mathcal{D}_t \, \alpha_t - \mathcal{D}_p \, \alpha_p\|_2^2, \tag{5}$$

where $\mathcal{D}_t$, $\mathcal{D}_p$ are the dictionary associated with the piecewise smooth component and textured component of the data, respectively. The final estimate is then found by adding together the two components obtained as $\hat{x}_p = \mathcal{D}_p \hat{\alpha}_p$ and $\hat{x}_t = \mathcal{D}_t \hat{\alpha}_t$. Note that, since the dictionaries are assumed to be tight frames, then $\mathcal{D}_p$ is the Moore-Penrose pseudo inverse of the analysis operator $\mathcal{W}_p$ associated with piecewise smooth data, i.e. $\mathcal{D}_p = \mathcal{W}_p^{\dagger}$ and, similarly, $\mathcal{D}_t$ is the Moore-Penrose pseudo inverse of the analysis operator $\mathcal{W}_t$ associated with texture data, i.e., $\mathcal{D}_t = \mathcal{W}_t^{\dagger}$.

The inpainting problem can be addressed within the framework of MCA as follows. Assume that the missing pixels are indicated by a diagonal mask matrix M where the main diagonal of M encodes the pixel status, namely 1 for an existing pixel and 0 for a missing one. Thus, in (5) we can incorporate this

[2] This last formulation is known in statistics as penalized least square estimation problem.

mask by [19]

$$\hat{\alpha}_t, \hat{\alpha}_p = \min_{\alpha_t, \alpha_p} \lambda \left(\|\alpha_t\|_1 + \|\alpha_p\|_1 \right) + \tfrac{1}{2} \, \|M(y - \mathcal{D}_t \, \alpha_t - \mathcal{D}_p \, \alpha_p)\|_2^2. \qquad (6)$$

The advantage of this approach is that the fidelity of the representation is measured with respect to the existing measurements only, disregarding missing pixels. Once $\mathcal{D}_t \, \alpha_t$ and $\mathcal{D}_p \, \alpha_p$ are recovered, those represent entire images, where holes are filled in by the two dictionaries' basis functions.

In this paper, rather than using a sparsity-based *synthesis model* as in (5), we prefer to use a sparsity-based *analysis model* leading to the minimization problem

$$\hat{x}_p, \hat{x}_t = \operatorname*{argmin}_{x_p, x_t} \lambda \|\mathcal{W}_p x_p\|_1 + \lambda \|\mathcal{W}_t x_t\|_1 + \tfrac{1}{2} \|M(y - x_p - x_t)\|_2^2 \qquad (7)$$

While in the synthesis formulation signals are modeled as sparse linear combinations of dictionary atoms, the analysis formulation emphasizes the zeros in the analysis side (rather than the non-zeros), leading to better performance. In particular, one of the major advantages of using the formulation (7) rather than (5) is that it requires searching lower dimensional vectors rather than longer dimensional representation coefficient vectors. To further improve the performance, we have also included a total variation regularization term, which is effective at reducing possible ringing artifacts near the edges. This idea appeared already in [17, 20]. Thus, we finally have the optimization problem:

$$\begin{aligned} \hat{x}_p, \hat{x}_t \quad &= \quad \arg \min_{x_p, x_t} \lambda \|\mathcal{W}_p \, x_p\|_1 + \lambda \|\mathcal{W}_t \, x_t\|_1 + \gamma \, TV(x_p) \\ & \qquad + \tfrac{1}{2} \|M(y - x_p - x_t)\|_2^2, \end{aligned} \qquad (8)$$

where TV is the Total Variation. The algorithm we use to solve this optimization problem is adapted from the algorithm of J. Starck et al [19, 21]. Once the separate estimates $\hat{x}_p$ and $\hat{x}_t$ are obtained as a solution of (8), the final estimator of x is $\hat{x} = \hat{x}_p + \hat{x}_t$.

2.3.1. Shearlet-based inpainting

The effectiveness of shearlets in the inpainting problem has been demonstrated by some recent theoretical and numerical papers [10, 22]. In our implementation of the inpainting scheme described above, we use a local discrete cosine dictionary for the representation of the texture component of the image (as dictionary $\mathcal{D}_t$) and a Parseval frame of shearlets for the representation of the piecewise smooth component of the image (as dictionary $\mathcal{D}_p$). Here we use the shearlet system constructed by one of the authors in [23], which combines simplicity of construction and numerical efficiency. As will be further discussed in Section 3, shearlets play a significant role in the overall performance of our denoising algorithm.

We recall that the shearlets are functions in $L^2(\mathbb{R}^2)$ obtained by applying a countable family of (anisotropic) dilations, translations and shear transformations to a finite set of generators. Hence, they form a collection of waveforms ranging not only at various scales and locations, like conventional wavelets, but also at various orientations and with highly anisotropic shapes [24, 25]. Due to these properties, it was proved that shearlets provide optimally sparse approximations for piecewise smooth images with edges [26], a property shared by curvelets but not by traditional wavelets. The approximation properties of shearlets stem in large part from their ability to represent very efficiently curvilinear edges and other types of distributed singularities [27, 28]. Thanks to these properties, one can *prove* that, when shearlets are applied to the solution of the inpainting problem according to the framework outlined above, then they are especially efficient to restore partially occluded curvilinear edges, significantly outperforming wavelets and other traditional methods [10]. In 4, we include a brief mathematical description of such properties. In the next section, we specifically illustrate the impact of the shearlet-based inpainting within our impulse noise denoising algorithm with respect to other more conventional inpainting methods.

3. Experimental results and discussion

In this section, we present extensive numerical experiments to illustrate the performance of our new impulse noise denoising algorithm and compare it against several state-of-the-art algorithms. We will denote our two-stage algorithm as *IIND+ST algorithm* to recall that it consists of our IIND impulse detection method followed by the shearlet-based inpainting described above. Note that numerical codes for the discrete shearlet decomposition are available at `www.math.uh.edu/∼dlabate` and `www.shearlab.org` (the latter website also contains the implementation of an inpainting routine similar to the one used here).

We have carried out our numerical experiments on the Lena, Boat and Barbara images, with size 512 × 512 pixels. For our tests, we have only used the Noise Models 3 and 4 described in the Section 2 to assess the performance of the algorithms. For the Noise Model 3, images were corrupted using random-valued impulse noise with different noise densities ranging from 10% to 80% and $m = 10$ (so that low intensity values are in the range 0-10 and high intensity values in the range 245-255); for Noise Model 4, we have also considered different noise densities and $m = 10$. This range of values is frequently assumed in the literature and is reasonable for a number of practical applications [1, 4]. As it is reasonable to expect, different images are affected in different ways by the noise and the restoration task is more challenging if the range of values of the image is not well separated by the range of values of this noise. This explains in large part the different denoising performance that we observed for different images and reported in Tables 2 and 3.

As objective quantitative measures of performance we have used the peak signal-to-noise ratio (PSNR) and the mean structural similarity (MSSIM) [29] between the original and restored images defined as

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{255^2}{\sum_i \sum_j (u_{i,j} - v_{i,j})^2 / MN} \right), \tag{9}$$

$$\mathrm{SSIM}(u_k, v_k) = \frac{(2\mu_{u_k}\mu_{v_k} + C_1)(2\sigma_{u_k v_k} + C_2)}{(\mu_{u_k}^2 + \mu_{v_k}^2 + C_1)(\sigma_{u_k}^2 + \sigma_{v_k}^2 + C_2)}, \tag{10}$$

$$\mathrm{MSSIM}(u, v) = \frac{\sum_{k=1}^{B} SSIM(u_k, v_k)}{B}, \tag{11}$$

where the symbols $u_{i,j}$ and $v_{i,j}$ denote the intensity values of the original image and the restored image, respectively; u_k and v_k are the image intensities at the k-th local window in the original and restored images; B is the total number of local windows; μ_l and σ_l, for $l = u_k, v_k$, are the mean and the standard deviation of u_k and v_k; $\sigma_{u_k v_k}$ is the covariance between u_k and v_k; $C_1 = (k_1 L)^2$ and $C_2 = (k_2 L)^2$ are small constants selected to stabilize SSIM using the following parameter settings: L is the dynamic range of the pixel values, $k_1 = 0.01$ and $k_2 = 0.03$. For measuring MSSIM, we have used the SSIM_index code by Wang et al. [29].

We have compared the performance of our algorithm against the conventional median filter (MED), the modified noise adaptive switching median filter (MNASM) [4] with HEIND detector [6] (H-MNASM), the switching median filter with boundary discriminative noise detection (SM-BDND) method [4], the fuzzy weighted non-local means (FWNLM) method [7] and the direction based adaptive weighted switching median (DAWSM) method [1]. Furthermore, to specifically illustrate the advantages of the shearlet-based inpainting within our RVIN denoising algorithm, we have also implemented a variant of our two-stage algorithm where the first stage is still our IIND impulse detection method, but the second stage uses a different inpainting method based on total variation. We will denote as *IIND+TV algorithm* this alternative two-stage denoising algorithms.

More precisely, the inpainting method used in our IIND+TV algorithm adopts the TV-H^{-1} method [30, 31], a successful and powerful restoration approach that aims to recover the missing pixels in an image by diffusing the information from the surrounding through using an appropriate evolution equation. That is, the inpainted image u is obtained from the damaged image $f \in L^2(\Omega)$, where $\Omega \in \mathbb{R}^2$ is a bounded set, by evolution according to the equation

$$u_t = -\triangle p + \lambda(f - u)\chi_{\Omega \setminus C}, \quad p \in \partial TV(u) \tag{12}$$

where $\partial TV(u)$ is the subdifferential of the total variation $TV(u)$, λ is a constant and C is the inpainting domain. We refer to [31] for additional details about the TV-H^{-1} inpainting method.

Tables 2 and 3 show the in-line performance comparison of the various image denoising methods considered. Note that the larger values the PSNR and MSSIM correspond to better quality of the restored image. The results in the tables show that our algorithms are extremely competitive and provide higher PSNR and MSSIM values than all other methods considered. In particular, the IIND+ST algorithm is the overall best performing method among all methods that we have considered, with improvements of several dBs. The performance of IIND+ST is outstanding at high noise levels where competing algorithms perform rather poorly and IIND+ST is able to outperform competeing methods by 10 dBs or more (see Table 3 for noise density at 0.8). To further illustrate the performance of our denoising approach we have also included some examples to provide a visual quality comparison among the various methods considered. Fig. 3 and Fig. 4 show the Lena and Boat images corrupted by RVIN of model 3 with 80% noise density; Fig. 5 and Fig. 6 show Lena and Boat images corrupted by RVIN of model 4 with 15% of low intensity impulse noise and 45% of high intensity impulse noise. The figures show that the methods MED, SM-BDND and FWNLM perform poorly for images that are corrupted by this high-level noise; the algorithms H-MNASM and DAWSM are somewhat more efficient to remove RVIN but produce some jagged edges and block artifacts as can be seen by inspecting Fig. 3(d, g), Fig. 4(d, g), Fig. 5(d, g) and Fig. 6(d, g). By contrast, our IIND+TV algorithm achieves a better visual performance when compared with the other five methods. The IIND+ST algorithm achieves the best visual performance overall and it restores the images without any significant artifacts such as block effect and jagged edges. Thus, both subjective and objective comparisons indicate that IIND+ST provides the best denoising results among all the methods considered. The improvement with respect to competing algorithms at high noise level is outstanding.

Fig. 7 shows the mean computation times of the various methods considered in this paper. All computations have been performed in Matlab software on a Centrino Duo personal computer, using a 2.00-GHz processor, running Windows 7. As it can be seen in Fig. 7, the mean computation time of IIND+ST is higher than IIND+TV. Both methods have higher computational cost than MED, H-MNASM, SM-BDND and DAWSM, but are less than FWNLM. Compared with DAWSM, for example, IIND+TV requires about twice the computation time and IIND+ST about 5 times the computation time. This shows that the improved performance comes at a rather significant increase in computational cost. The performance table indicates that this additional cost is certainly worth the effort for higher noise levels. In fact, when noise density level is 0.8, IIND+ST outperforms DAWSM by at least 5 dBs and up to 19 dBs (in the case of the Lena image, for Noise Model 4). For lower noise levels the improvement is more modest (between 2.5 and 6 dBs), and the user will have to evaluate the trade off between computational cost and denoising performance.

On the other hand, it is possible to further speed up the implementation of the IIND+ST algorithm by taking advantage of recent faster numerical implementations of the shearlet transform. Recent results by one of the authors and collaborators show that the shearlet transform implementation can be parallelized and, by taking advantage of GPUs, the computation time of the shearlet transform can be reduced by up to 50 times [33]. Since this is the slowest part of the IIND+ST code, this improvement would significantly reduce the overall computational cost of the algorithm.

4. Conclusion

In this paper, we introduced a new, efficient two-stage denoising algorithm to restore images corrupted by RVIN. Our approach includes an improved routine for impulse noise detection that is a refinement of the Highly Effective Impulse Noise Detection (HEIND) algorithm and an innovative image inpainting routine that is designed to recover the corrupted pixel while efficiently preserving the geometrical content of the original image. This inpainting routine takes advantage of the principle of morphological component analysis and the power of the shearlet representation.

We illustrated the performance of our approach using images corrupted by two classical types of random-valued impulse noise with a wide range of noise levels, ranging from 10% to 80%. To assess the performance of our approach, we compared it against several state-of-the-art random-valued impulse noise denoising methods. The experimental results reported in this paper show that our method is extremely efficient in removing RVIN and recovering the lost information even for high noise levels, where the improvement with respect to competing methods is often outstanding. Our two-stage method is highly competitive in terms of both objective measures and visual quality and outperforms similar two-stage methods with a modest increase in computational cost.

Appendix A. Shearlet-based inpainting

We recall a few facts from the asymptotic analysis of inpainting using shearlets (cf. [10, 32].

Since we will focus on the problem of restoring images with edges, let us consider the following distributional model. Let $w \in C^\infty(\mathbb{R}^2)$ be a window function such that $0 \leq w \leq 1$ and supp $w \subset [-R, R]$ for some $R > 0$. Hence we define the (weighted) *distributional edge* $\mathcal{L}_w$ by

$$\langle \mathcal{L}_w, \phi \rangle = \int_{-R}^{R} w(x_1)\, \phi(x_1, 0)\, dx_1,$$

where $\phi \in \mathcal{SH}(\mathbb{R}^2)$ is a Schwartz class function. This definition models an horizontal edge with support in $[-R, R] \times \{0\}$.

To model the data loss, we define the mask

$$M_h = \{(x_1, x_2) \in \mathbb{R}^2 : |x_1| \leq h\},$$

where $h > 0$ is the gap size. Hence we can break up the image space as $L^2(\mathbb{R}^2) = L^2(M_h) \bigoplus L^2(M_h^c)$ where $L^2(M_h)$ is the space of functions in $L^2(\mathbb{R}^2)$ with support in M_h and similarly for $L^2(M_h^c)$. Thus, the inpainting problem can be formulate as the problem of *recovering the image* $\mathcal{L}_w$ *given the corrupted image* $P_{M_h^c}\mathcal{L}_w$, where $P_{M_h^c}$ denotes the orthogonal projection of $L^2(\mathbb{R}^2)$ into

$L^2(M_h^c)$. The interesting question is: what is the largest gap h that can be recovered?

To tackle this problem, we start by decomposing $\mathcal{L}_w$ into multiple subbands as

$$(\mathcal{L}_w)_j = \mathcal{L}_w * F_j, \quad j \geq 0$$

using appropriate filters F_j. More precisely, the filters are defined in the Fourier domain as $\widehat{F}_j(\xi) = W(2^{2j}\xi)$ where $W(\xi) = \sqrt{\widehat{\Phi}^2(2^{-2}\xi) - \widehat{\Phi}^2(\xi)}$ and $\widehat{\Phi}(\xi) = \widehat{\Phi}(\xi_1, \xi_2) = \hat{\phi}(\xi_1)\,\hat{\phi}(\xi_2)$ and ϕ is a C^∞ univariate function such that $0 \leq \hat{\phi} \leq 1$, $\hat{\phi} = 1$ on $[-\frac{1}{16}, \frac{1}{16}]$ and $\hat{\phi} = 0$ outside the interval $[-\frac{1}{8}, \frac{1}{8}]$. Note that $\widehat{F}_j(\xi)$ is a C^∞ function with support inside the Cartesian corona $[-2^{2j-1}, 2^{2j-1}]^2 \setminus [-2^{2j-4}, 2^{2j-4}]^2$.

Now, we set the ℓ_1 minimization problem

$$L_j = \arg \min_{L \in L^2(\mathbb{R}^2)} \|\mathcal{W}L\|_{\ell_1} \quad \text{subject to } P_{M_h^c} L = P_{M_h^c} \mathcal{L}_w, \tag{13}$$

where $\mathcal{W}$ is the analysis operator associated with the shearlet decomposition (as in Section 2.3). We have the following remarkable result from [10].

Theorem. For $h = h_j = o(2^{-j/2})$ and L_j the solution of (13), we have that

$$\frac{\|L_j - (\mathcal{L}_w)_j\|_{L^2}}{\|(\mathcal{L}_w)_j\|_{L^2}} \to 0, \quad j \to \infty.$$

In other words, we have asymptotically perfect inpainting if the size of the gap shrinks faster than $2^{-j/2}$. This is, in a precise sense, the best result achievable and follows from the properties of the shearlet representation. Wavelets, by comparison, can only successfully inpaint gaps that are smaller than 2^{-j} (cf. [10] for more details).

Acknowledgment

Guorong Gao is grateful for the hospitality of the Department of Mathematics of the University of Houston where this work was completed. Demetrio Labate acknowledges support from grants NSF-DMS 1008900, 100579, 1320910.

References

[1] M. S. Nair and P. M. Ameera Mol, "Direction based adaptive weighted switching median filter for removing high density impulse noise", *Comput Electr Eng*, vol. 39, no. 2, pp. 663–689, Feb.2013.

[2] C. T. Lu and T. C. Chou, "Denoising of salt-and-pepper noise corrupted image using modified directional-weighted-median filter", *Pattern Recognition Letters*, vol. 33, no. 10, pp. 1287–1295, Jul.2012.

[3] Y. Dong and S. Xu, “A new directional weighted median filter for removal of random-valued impulse noise”, *IEEE Signal Proc Lett*, vol. 14, no. 3, pp. 193–196, Nov.2007.

[4] P. E. Ng and K. K. Ma, “A switching median filter with boundary discriminative noise detection for extremely corrupted images”, *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1506–1516, Jun.2006.

[5] J. Wu and C. Tang, “A new filter for the removal of random-valued impulse noise from highly corrupted images”, *Int. J. Electron. Commun.*, vol. 66, no. 10, pp. 847–854, Oct.2012.

[6] F. Duan and Y. J. Zhang, “A highly effective impulse noise detection algorithm for switching median filters”, *IEEE Signal Processing Letters*, vol. 17, no. 7, pp. 647–650, Jul.2010.

[7] J. Wu and C. Tang, “Random-valued impulse noise removal using fuzzy weighted non-local means”, *SIViP*, vol. 8, no. 2, pp. 349–355, Feb.2014.

[8] G. Gao and Y. Liu, “An efficient three-stage approach for removing salt and pepper noise from digital images”, *International Journal for Light and Electron Optics*, vol. 126, no. 4, pp. 467–471, 2015

[9] J. Wu and C. Tang, “An efficient decision-based and edge-preserving method for salt-and-pepper noise removal”, *Pattern Recognition Letters*, vol. 32, no. 15, pp. 1974–1981, Nov.2011.

[10] E. J. King, G. Kutyniok and X. S. Zhuang, “Analysis of inpainting via clustered sparsity and microlocal analysis”, *Journal of Mathematical Imaging and Vision*, vol. 48, no. 2, pp. 205–234, Feb.2014.

[11] K. K. Anisha and M. Wilscy, “Impulse noise removal from medical images using fuzzy genetic algorithm,” *Signal and Image Processing: an International Journal* (SIPIJ), vol. 2, no. 4, pp. 173-186, 2011.

[12] M. H. Hsieh, F. C. Cheng, M. C. Shie and S. J. Ruan, “Fast and efficient median filter for removing $1 - 99\%$ levels of salt-and-pepper noise in images”, *Eng. Appl. Artif. Intel.*, vol. 26, no. 4, pp. 1333–1338, Apr.2013.

[13] T. Chan, J. Shen, “Local inpainting models and TV inpainting”, *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, 2001

[14] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, J. Verdera, “Filling-in by joint interpolation of vector fields and grey levels”, *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1200–1211, 2001.

[15] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit”, *SIAM Rev.*, vol. 43, no. 1, pp. 125–159, 2001.

[16] G. Easley, D. Labate and P.S. Negi, "3D data denoising using combined sparse dictionaries", *Math. Model. Nat. Phen.*, vol. 8, no. 1, pp. 60–74, 2013.

[17] J.-L. Starck, M. Elad, and D.L. Donoho, "Image Decomposition Via the Combination of Sparse Representation and a Variational Approach", *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1570–1582, Oct.2005.

[18] D. Donoho, and G. Kutyniok, "Microlocal analysis of the geometric separation problem", *Comm. Pure Appl. Math.*, vol. 66, no. 1, pp. 1–47, Jan.2013.

[19] M. Elad, J.-L Starck, D. Donoho and P. Querre, "Simultaneous Cartoon and Texture Image Inpainting using Morphological Component Analysis (MCA)", *Applied Computat. Harmon. Anal.*, vol. 19, no. 3, pp. 340–358, Nov.2005.

[20] F. Malgouyres, "Minimizing the total variation under a general convex constraint for image restoration", *IEEE Trans. Signal Process.*, vol. 11, no. 12, pp. 1450–1456, 2002.

[21] J. Starck, M. Elad and D.L. Donoho, "Image decomposition via the combination of sparse representation and a variational approach", *IEEE Trans. Image Proc.*, vol. 14, no. 10, pp. 1570–1582, 2005.

[22] M. Genzel and G. Kutyniok, "Asymptotic Analysis of Inpainting via Universal Shearlet Systems", *SIAM J. Imaging Sci.*, to appear.

[23] G. Easley, D. Labate and W. Lim, "Sparse directional image representations using the discrete shearlet transform", *Appl. Comput. Harmon. Anal.*, vol. 25, pp. 25-46, 2008.

[24] G. Kutyniok and D. Labate, Shearlets: Multiscale analysis for multivariate data, Birkhäuser/Springer, New York, Boston, 2012.

[25] D. Labate, W. Lim, G. Kutyniok, and G. Weiss, Sparse multidimensional representation using shearlets, Wavelets XI (San Diego, CA, 2005), 254–262, SPIE Proc. 5914, SPIE, Bellingham, WA, 2005.

[26] K. Guo and D. Labate, "Optimally sparse multidimensional representation using shearlets", *SIAM J. Math. Anal.*, vol. 39, no. 1, pp. 298–318, 2007.

[27] K. Guo, and D. Labate, "Characterization and analysis of edges using the continuous shearlet transform", *SIAM Journal on Imaging Sciences*, vol. 2, no. 3, pp. 959–986, 2009.

[28] K. Guo, and D. Labate, "Characterization of piecewise-smooth surfaces using the 3D continuous shearlet transform", *J. Fourier Anal. Appl.*, vol. 18, pp. 488–516, 2012.

[29] Z. Wang, A. C. Bovik, H. R. Sheik and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity”, *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr.2004.

[30] C. Schönlieb and A. Bertozzi, “Unconditionally stable schemes for higher order inpainting”, *Commun. Math. Sci.*, vol. 9, no. 2, pp. 413–457, 2011.

[31] M. Burger, L. He and C. Schönlieb, “Cahn-Hilliard inpainting and a generalization for grayvalue images”, *SIAM J. Imaging Sci.*, vol. 2, no. 4, pp. 1129–1167, 2009.

[32] K. Guo, and D. Labate, “The construction of smooth Parseval frames of shearlets”, *Math. Model. Nat. Phenom.*, vol. 8, no. 1, pp. 82–105, 2013.

[33] X. G. Serra, V. M. Patel, D. Labate and R. Chellappa, “Discrete Shearlet Transform on GPU with applications in anomaly detection and denoising”, *EURASIP J. Adv. Signal Process*, vol. 64 (10 May 2014), 2014.

Figures and Tables

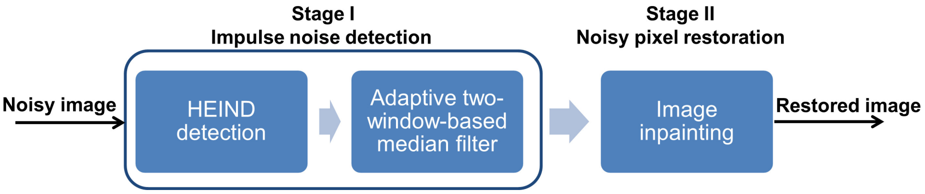


Fig. 1. Flowchart of the proposed two-stage RVIN denoising algorithm.

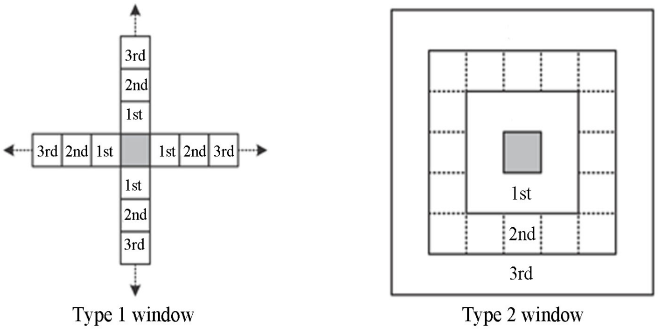
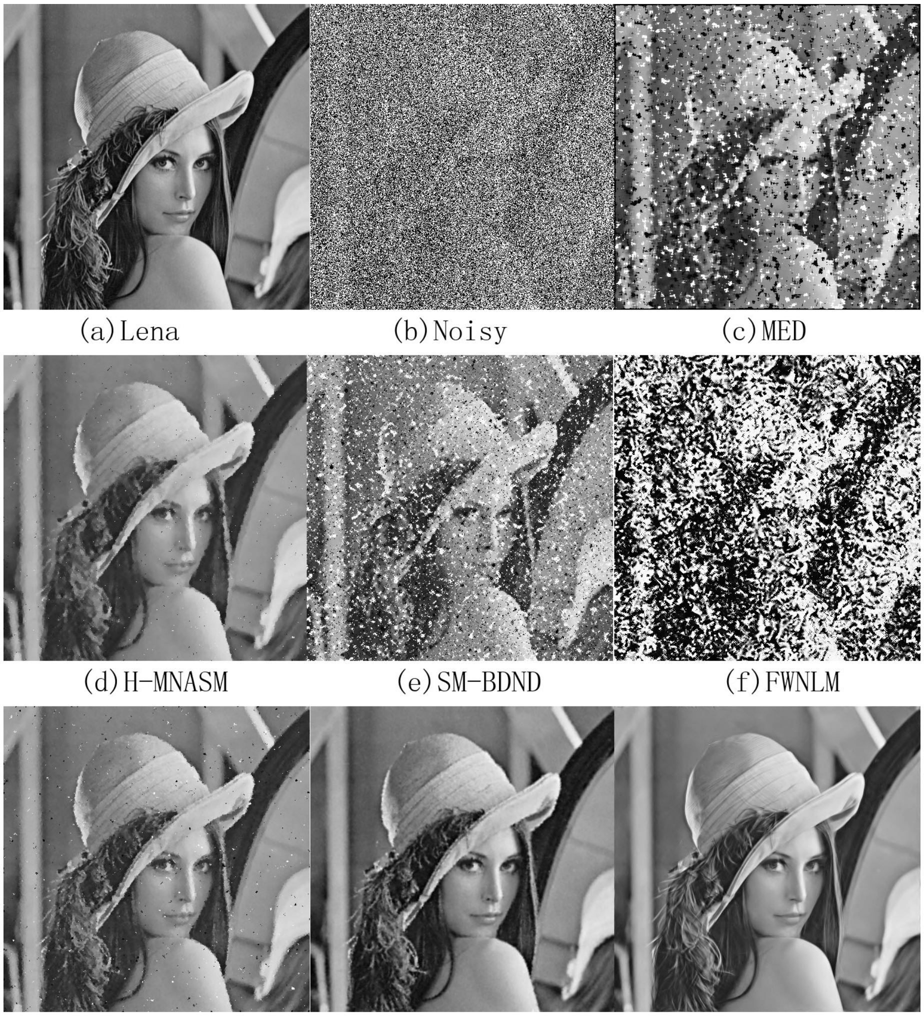


Fig. 2. Type 1 and Type 2 windows used in the two-window based median filter.

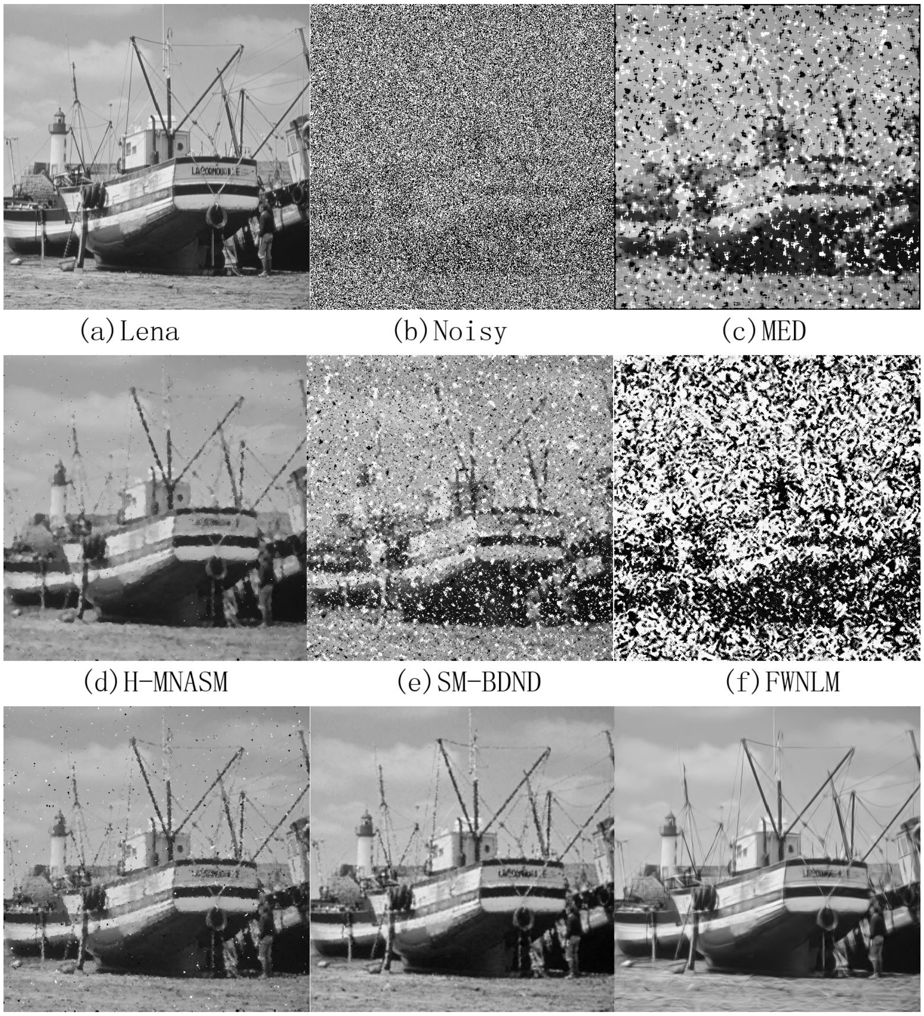


(a)Lena (b)Noisy (c)MED

(d)H-MNASM (e)SM-BDND (f)FWNLM

(g)DAWSM (h)IIND+TV (i)IIND+ST

Fig. 3. Restoration of Lena image corrupted by RVIN, Model 3, with 80% noise density and $m = 10$.

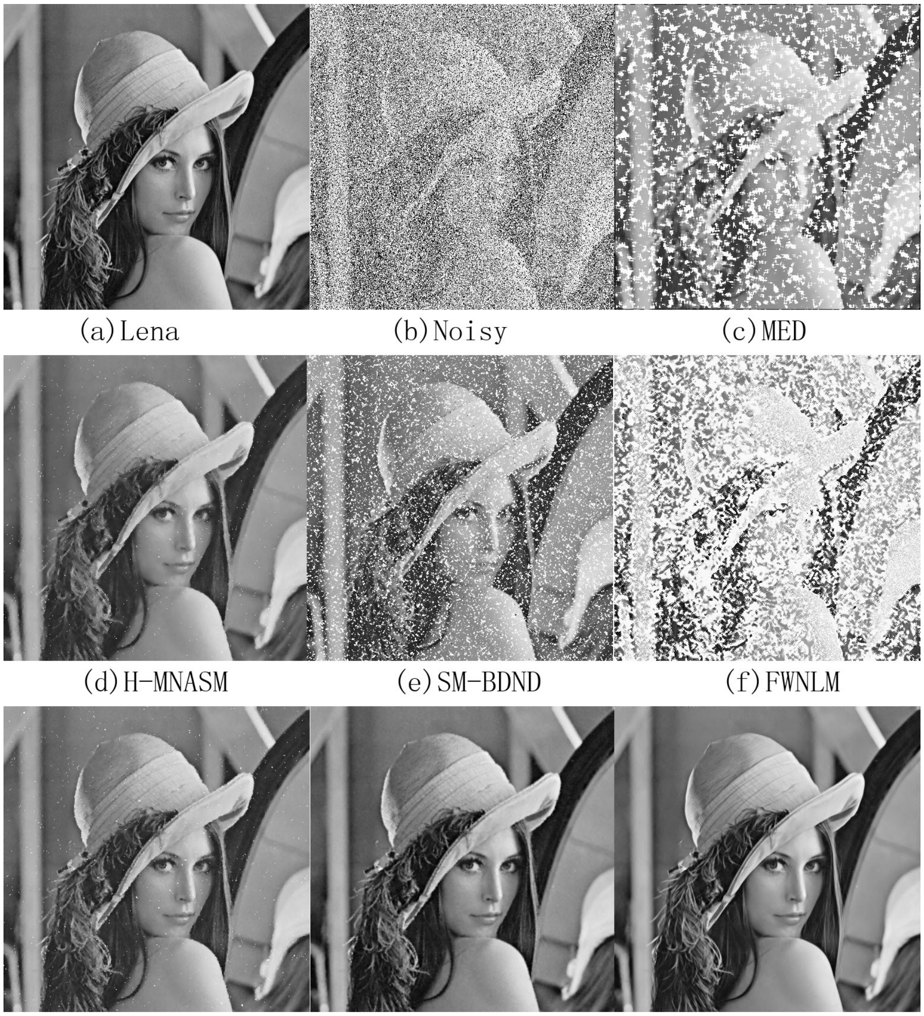


(a)Lena (b)Noisy (c)MED

(d)H-MNASM (e)SM-BDND (f)FWNLM

(g)DAWSM (h)IIND+TV (i)IIND+ST

Fig. 4. Restoration of Boat image corrupted by RVIN, Model 3, with 80% noise density and $m = 10$.

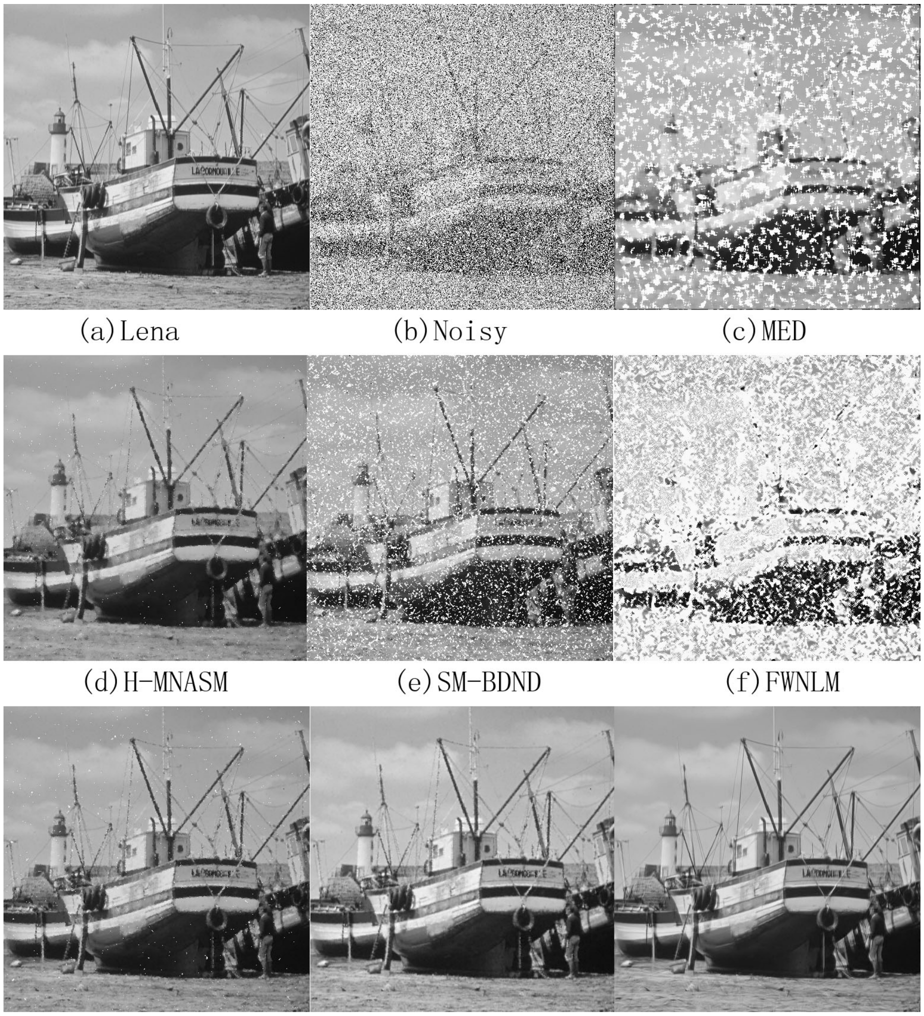


(a) Lena (b) Noisy (c) MED

(d) H-MNASM (e) SM-BDND (f) FWNLM

(g) DAWSM (h) IIND+TV (i) IIND+ST

Fig. 5. Restoration of Lena image corrupted by RVIN, Model 4, with 60% noise density and $m = 10, p_1 = 15\%, p_2 = 45\%$.


(a)Lena (b)Noisy (c)MED

(d)H-MNASM (e)SM-BDND (f)FWNLM

(g)DAWSM (h)IIND+TV (i)IIND+ST

Fig. 6. Restoration of Boat image corrupted by RVIN, Model 4, with 60% noise density and $m = 10, p_1 = 15\%, p_2 = 45\%$.

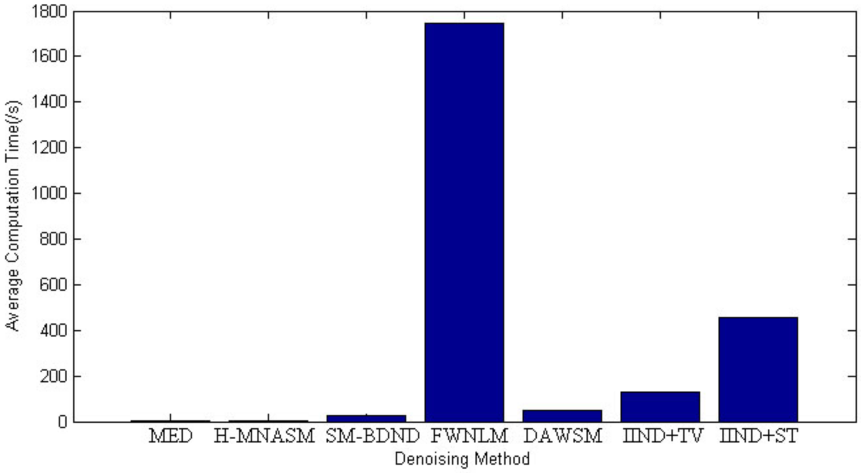


Fig. 7. Mean computation times using different image restoration algorithms.

Image	Noise model	Noise density	HEIND	IIND
Lena	Model 3	0.1	1	1
		0.2	0	0
		0.3	1	1
		0.4	20	0
		0.5	27	1
		0.6	115	0
		0.7	268	0
		0.8	625	0
	Model 4	0.1	1	0
		0.2	-1	-1
		0.3	13	-1
		0.4	62	-2
		0.5	208	-1
		0.6	644	0
		0.7	4756	-3
		0.8	24644	12
Boat	Model 3	0.1	2	2
		0.2	0	0
		0.3	10	0
		0.4	19	0
		0.5	64	0
		0.6	125	0
		0.7	331	0
		0.8	688	3
	Model 4	0.1	114	-11
		0.2	149	6
		0.3	111	-12
		0.4	105	-21
		0.5	203	-17
		0.6	563	-8
		0.7	4568	-24
		0.8	24053	6

Tab. 1. The table compares the difference between the number of true noisy pixels and the number of noisy pixels detected using HEIND and IIND routines. The noise is generated using Model 3, with $m = 10$, and Model 4, with $m = 15$. We set $p_1 = \frac{p}{4}, p_2 = \frac{3p}{4}$, where p is the noise density.

Tab. 2. Restoration results in PSNR and MSSIM on images corrupted by RVIN of Model 3.

Image	Index	Noise	MED	H-MNASM	SM-BDND	FWNLM	DAWSM	IIND+TV-H	IIND+ST
Lena	PSNR	0.1	28.49	42.78	42.88	35.58	42.73	43.79	45.09
		0.2	27.09	39.20	36.52	33.11	39.27	40.71	42.13
		0.3	25.73	36.46	30.54	30.00	36.72	38.45	39.98
		0.4	24.71	33.96	26.02	25.23	34.84	36.85	38.43
		0.5	23.78	31.62	21.98	19.59	32.93	35.23	36.84
		0.6	22.26	29.94	18.86	14.16	31.02	33.75	35.45
		0.7	18.4	28.09	16.09	9.96	28.68	32.02	33.83
		0.8	13.51	26.21	13.23	7.61	24.87	30.18	31.75
	MSSIM	0.1	0.8144	0.9898	0.9899	0.9185	0.98984	0.99121	0.99157
		0.2	0.8090	0.9787	0.9695	0.9026	0.97891	0.98235	0.98346
		0.3	0.8027	0.9644	0.9146	0.8655	0.96536	0.97164	0.97444
		0.4	0.7924	0.9438	0.7974	0.7590	0.94971	0.95992	0.96494
		0.5	0.7773	0.9161	0.6037	0.5093	0.92895	0.94531	0.95298
		0.6	0.7210	0.8821	0.3922	0.1933	0.89947	0.92729	0.94013
		0.7	0.5338	0.8347	0.2252	0.0636	0.85452	0.90369	0.9238
		0.8	0.2202	0.7703	0.1164	0.0293	0.75118	0.86705	0.89621
Boat	PSNR	0.1	25.71	39.77	39.96	32.44	40.11	41.72	43.66
		0.2	24.69	36.30	34.76	30.15	36.40	38.30	40.31
		0.3	23.60	33.53	29.81	27.39	34.11	36.16	37.94
		0.4	22.77	31.12	25.26	23.99	32.06	34.27	36.06
		0.5	21.82	29.15	21.70	18.81	30.48	32.70	34.44
		0.6	20.63	27.30	18.67	13.71	28.81	31.13	32.76
		0.7	17.59	25.85	16.00	9.88	26.85	29.60	30.96
		0.8	13.01	24.25	13.17	7.46	23.90	27.88	29.09
	MSSIM	0.1	0.7408	0.9880	0.9888	0.9255	0.98891	0.99144	0.99298
		0.2	0.7352	0.9753	0.9686	0.8929	0.97575	0.98181	0.98552
		0.3	0.7280	0.9562	0.9108	0.8387	0.95989	0.97028	0.97646
		0.4	0.7177	0.9299	0.7976	0.7278	0.93989	0.95613	0.96678
		0.5	0.6997	0.8957	0.6136	0.4780	0.91575	0.93848	0.95444
		0.6	0.6507	0.8466	0.4138	0.1923	0.88179	0.91539	0.93775
		0.7	0.4838	0.7912	0.2379	0.0760	0.83139	0.88474	0.91361
		0.8	0.2030	0.7179	0.1239	0.0345	0.73027	0.83859	0.87795
Barbara	PSNR	0.1	23.13	33.72	33.71	28.90	33.73	34.70	43.44
		0.2	22.64	30.40	29.93	26.45	30.43	31.47	39.88
		0.3	22.05	28.33	26.75	24.09	28.43	29.41	37.67
		0.4	21.55	26.84	23.59	21.18	27.02	27.97	35.74
		0.5	20.93	25.73	20.68	17.08	25.76	26.68	34.04
		0.6	19.88	24.67	17.82	13.00	24.46	25.54	31.83
		0.7	17.06	23.65	15.34	9.67	23.22	24.45	29.50
		0.8	12.71	22.41	12.41	7.36	21.34	23.38	26.06
	MSSIM	0.1	0.65131	0.98083	0.98084	0.91838	0.98085	0.98374	0.99375
		0.2	0.6464	0.95716	0.95261	0.87181	0.95761	0.96446	0.98666
		0.3	0.63918	0.92926	0.89258	0.79172	0.93144	0.94204	0.9789
		0.4	0.63028	0.89445	0.77457	0.64701	0.90148	0.91619	0.96958
		0.5	0.61471	0.85508	0.61567	0.42912	0.86542	0.88582	0.95791
		0.6	0.57536	0.80378	0.41587	0.18966	0.81465	0.84626	0.93819
		0.7	0.43476	0.74262	0.25679	0.071393	0.75194	0.80096	0.90822
		0.8	0.18377	0.66162	0.13388	0.028673	0.64356	0.73547	0.83195

Tab. 3. Restoration results in PSNR and MSSIM on images corrupted by RVIN of Model 4 ($p_1 = \frac{p}{4}, p_2 = \frac{3p}{4}$).

Image	Index	Noise	MED	H-MNASM	SM-BDND	FWNLM	DAWSM	IIND+TV-H	IIND+ST
Lena	PSNR	0.1	28.67	42.70	41.72	35.57	42.79	43.81	44.99
		0.2	27.93	39.02	32.55	31.92	39.01	40.64	42.10
		0.3	26.43	36.27	25.60	25.38	36.69	38.43	40.06
		0.4	23.90	33.66	20.57	17.96	34.21	36.75	38.32
		0.5	18.10	30.83	16.69	12.30	31.57	35.15	36.89
		0.6	11.67	27.13	13.58	8.34	28.56	33.79	35.46
		0.7	7.76	17.49	10.44	6.41	21.43	31.97	33.84
		0.8	6.14	10.63	7.63	5.75	12.28	28.68	31.78
	MSSIM	0.1	0.8133	0.9903	0.9891	0.9184	0.98977	0.99111	0.99148
		0.2	0.8056	0.9783	0.9437	0.8939	0.97838	0.98226	0.98346
		0.3	0.7906	0.9632	0.8010	0.7729	0.96476	0.97125	0.97409
		0.4	0.7515	0.9407	0.5693	0.4424	0.94553	0.9594	0.96464
		0.5	0.5505	0.9022	0.3294	0.1554	0.91414	0.94499	0.95316
		0.6	0.2125	0.8167	0.1791	0.0723	0.85203	0.9277	0.94048
		0.7	0.1068	0.3142	0.0895	0.1890	0.64213	0.89886	0.9229
		0.8	0.2641	0.0973	0.0538	0.1378	0.15073	0.81723	0.89715
Boat	PSNR	0.1	25.75	39.72	39.15	32.44	39.88	41.60	43.56
		0.2	24.94	36.13	32.05	27.27	36.47	38.34	40.29
		0.3	23.75	33.57	25.80	23.82	33.85	35.99	37.95
		0.4	22.07	31.01	21.08	17.44	31.93	34.27	36.05
		0.5	17.71	28.78	17.25	12.28	29.87	32.53	34.24
		0.6	11.98	26.02	14.14	9.01	27.94	31.12	32.76
		0.7	8.27	17.57	10.99	7.17	23.89	29.55	31.05
		0.8	6.75	11.02	8.21	6.43	12.68	26.76	28.84
	MSSIM	0.1	0.7376	0.9880	0.9878	0.9250	0.98872	0.99149	0.99298
		0.2	0.7252	0.9748	0.9462	0.8484	0.97563	0.98158	0.98552
		0.3	0.7068	0.9571	0.8210	0.7417	0.95863	0.97011	0.9771
		0.4	0.6668	0.9279	.6016	0.4188	0.93618	0.95591	0.96636
		0.5	0.4952	0.8830	0.3799	0.1394	0.90489	0.93769	0.95396
		0.6	0.1944	0.7996	0.2160	0.0643	0.84282	0.91611	0.93884
		0.7	0.1071	0.3321	0.1134	0.1770	0.64963	0.87962	0.91372
		0.8	0.2617	0.1144	0.0686	0.1359	0.17162	0.78813	0.87646
Barbara	PSNR	0.1	23.16	33.74	33.60	28.98	33.74	34.76	43.64
		0.2	22.75	30.51	28.70	25.89	30.55	31.59	40.04
		0.3	22.00	28.34	23.86	21.52	28.48	29.49	37.67
		0.4	20.69	26.76	19.53	15.83	26.94	27.97	35.74
		0.5	16.63	25.32	15.92	10.88	25.31	26.59	33.80
		0.6	10.89	23.91	12.91	7.65	23.64	25.50	31.90
		0.7	7.12	20.61	10.06	5.91	19.55	24.31	29.46
		0.8	5.56	12.68	7.15	5.24	11.36	21.06	25.94
	MSSIM	0.1	0.65121	0.98072	0.97949	0.91835	0.98073	0.98375	0.99375
		0.2	0.64137	0.95791	0.93276	0.86306	0.95841	0.96526	0.98685
		0.3	0.62809	0.92835	0.79751	0.71265	0.93097	0.94248	0.97898
		0.4	0.5949	0.89143	0.58051	0.41609	0.89842	0.91618	0.96958
		0.5	0.43686	0.83942	0.37653	0.16014	0.84922	0.88403	0.95709
		0.6	0.17787	0.76638	0.22185	0.071622	0.77663	0.84453	0.93898
		0.7	0.084802	0.58551	0.12706	0.12893	0.58317	0.78033	0.90756
		0.8	0.17731	0.18324	0.072671	0.22409	0.15419	0.5709	0.82726