

Mixed Noise Removal by Weighted Encoding with Sparse Nonlocal Regularization

Jielin Jiang, Lei Zhang, and Jian Yang

Abstract—Mixed noise removal from natural images is a challenging task since the noise distribution usually does not have a parametric model and has a heavy tail. One typical kind of mixed noise is additive white Gaussian noise (AWGN) coupled with impulse noise (IN). Many mixed noise removal methods are detection based methods. They first detect the locations of impulse noise pixels and then remove the mixed noise. However, such methods tend to generate many artifacts when the mixed noise is strong. In this paper, we propose a simple yet effective method, namely weighted encoding with sparse nonlocal regularization (WESNR), for mixed noise removal. In WESNR, there is not an explicit step of impulse pixel detection; instead, soft impulse pixel detection via weighted encoding is used to deal with IN and AWGN simultaneously. Meanwhile, the image sparsity prior and nonlocal self-similarity prior are integrated into a regularization term and introduced into the variational encoding framework. Experimental results show that the proposed WESNR method achieves leading mixed noise removal performance in terms of both quantitative measures and visual quality.

Index Terms—Mixed noise removal, weighted encoding, non-local, sparse representation.

I. INTRODUCTION

DURING image acquisition and/or transmission, noise will be more or less introduced. Denoising (or noise removal) is a fundamental problem in image processing, aiming to estimate the original image from its noise-corrupted observation while preserving as much as possible the image edges, textures and fine scale details. The prior knowledge of noise distribution plays an important role in noise removal. Two types of commonly encountered noise are additive white Gaussian noise (AWGN) and impulse noise (IN). AWGN is often introduced due to the thermal motion of electron in camera sensors and circuits [22]. IN is often introduced by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or bit errors in transmission [23]. Many papers have been published on removing either AWGN [12]-[20] or IN [2]-[11]. The mixture of AWGN and IN, however, is also commonly encountered in practice due to the multiple sources of noise. A variety of mixed noise removal methods have been proposed in past decades [22]-[36].

An image corrupted by IN will have a portion of its pixels replaced by random noise values with the remaining pixels unchanged. Two types of widely encountered IN are salt-and-pepper impulse noise (SPIN) and random-valued impulse noise

(RVIN). An image corrupted by SPIN shows dark pixels in bright regions and bright pixels in dark regions. Nonlinear filters such as median filters [1] have been dominantly used to remove IN. However, one shortcoming of median filters is that the image local structures can be destroyed, making the denoised images look unnatural. This problem becomes serious when the IN density is high. Various improvements of median filters have been proposed to better preserve the image local structures [2]-[10]. Among them, the weighted median filter [2], the center-weighted median filter [3] and the multistate median filter [4] do not distinguish whether the current pixel is a noise pixel or not, and they tend to over-smooth the fine scale image details. An alternative way is to detect and process the corrupted IN pixels, and leave the uncorrupted pixels unchanged. The representative methods along this line include switching median filter [5], adaptive median filter (AMF) [6], tristate median filter [7], adaptive center-weighted median filter [8], conditional signal-adaptive median filter [9], and directional weighted median filter [10], etc. The genetic programming filter [11] by switching between two IN detectors and their associated estimators was also developed for IN removal.

AWGN is the most widely studied noise model in image denoising literature [12]-[20]. At each pixel of an image corrupted by AWGN, a value independently sampled from a zero-mean Gaussian distribution is added to the pixel gray level. Traditional linear filtering methods such as Gaussian filtering can smooth noise efficiently but they will over-smooth the image edges at the same time. To solve this problem, nonlinear filtering methods have been developed. The well-known bilateral filter (BF) [12] is good at edge preservation. It estimates each pixel as the weighted average of the neighboring pixels but the weights are determined by both the intensity similarity and spatial similarity. The nonlocal means (NLM) filtering method [15] can be viewed as a significant extension of BF based on the fact that similar pixels in an image can be spatially far from each other. In NLM, each pixel is estimated as the weighted average of all its similar pixels in the image, and the weights are determined by the similarity between them. By grouping the nonlocal similar patches into a 3D cube and applying transform based shrinkage, the BM3D method [14] has become a benchmark for AWGN removal. Zhang et al. [16] grouped the similar patches into a matrix and applied principal component analysis (PCA) to remove AWGN. The so-called LPG-PCA algorithm achieves very good edge preservation performance. In recent years, the sparse representation and dictionary learning based methods have been attracting significant attention in image restoration. The seminal work of

J. Jiang and J. Yang are with the Department of Computer Science, Nanjing University of Science and Technology, Nanjing, China

L. Zhang is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong.

Corresponding author. Email: cszhang@comp.polyu.edu.hk.

K-SVD [13] initiates the study of learning a dictionary from natural images for AWGN removal. The joint use of sparse representation and nonlocal self-similarity regularization has lead to state-of-the-art AWGN removal performance [17]. Very recently, deep convolutional neural networks [18] have also shown powerful capability to remove AWGN.

The mixture of IN and AWGN, however, makes the denoising problem much more difficult because of the very different properties of the two types of noises. A few methods have been developed to remove the mixed IN and AWGN noise [22]-[36]. The median-based signal-dependent rank ordered mean (SDROM) filter [24] can be used for IN removal as well as mixed noise removal. However, when applied to image with mixed noise, it often produces visually unpleasant artifacts. The trilateral filter (TF) [27] incorporates the rank-order absolute difference (ROAD) statistics into the BF [12] framework for IN detection. Switching bilateral filter (SBF) [28] is also a modification of BF based on the method of detection and replacement. The reference median is computed to decide whether a current pixel is a noise pixel or not. If the absolute value between the reference median and a target pixel is large, then the target pixel is considered as a noise pixel, and consequently the mixed noise is removed by switching between the AWGN removal and IN removal. The FIRDM filter [29] contains two separate steps: an IN detection step and a noise reduction step that preserves edge sharpness. It can effectively remove SPIN, but its performance in removing RVIN is not satisfactory because RVIN may not produce large gradient values. The HDIR filter [26] removes mixed noise by kernel regression with Bayesian classification of the input pixels. In [30], a new IN detection mechanism based on robust outlyingness ratio (ROR) and NLM is proposed, where the image pixels are divided into four clusters according to the ROR value and by using an iterative coarse-to-fine strategy.

Cai et al. [31] proposed a modified two-phase method to reconstruct images corrupted by IN and AWGN mixed noise, and the efficiency of this method is improved in [32]. Xiao et al. [33] proposed an $l_1 - l_0$ minimization approach to mixed noise removal. This method achieves state-of-the-art denoising results but its computational complexity is somewhat high. Rodríguez et al. [34] proposed a cost functional consisting of a TV regularization term and l_2 and l_1 data fidelity terms, which aim to reduce AWGN and IN, respectively. This method achieves competitive mixed noise removal results but with much better computational performance. Dong et al. [35] presented two sparsity-based regularization models for blind inpainting problems. A new variable is introduced in the data fidelity term to represent the outliers. Meanwhile, this new variable is used as a regularizer by assuming that the percentage of pixels damaged by IN is small. Recently, Liu et al. [36] proposed a weighted dictionary learning model for mixed noise removal. This method integrates sparse coding and dictionary learning, image reconstruction, noise clustering and parameters estimation into a four-step framework, and each step solves a minimization problem.

Many existing mixed noise removal methods are detection based methods and they involve two sequential steps, i.e., first detect the IN pixels and then remove the noise. Such a two-

phase strategy will become less effective when the AWGN or IN is strong. In this paper, we propose a simple yet effective encoding based method for mixed noise removal, namely weighted encoding with sparse nonlocal regularization (WESNR). There is no explicit impulse pixel detection in WESNR, and we encode each noise-corrupted patch over a pre-learned dictionary to remove the IN and AWGN simultaneously in a soft impulse pixel detection manner. The major difficulty of IN and AWGN mixed noise removal lies in the complex distribution of mixed noise, which has a heavy tail and cannot be readily characterized by a parametric model. The conventional l_2 -norm data fidelity term, which is well suited to characterize the Gaussian distributed data fitting residual, is not suitable to suppress the mixed noise with complex non-Gaussian distribution. In WESNR, the mixed noise is suppressed by weighting the encoding residual so that the final encoding residual will tend to follow Gaussian distribution. The weighted encoding and sparse nonlocal regularization are unified into a variational framework, which is easy to minimize. Extensive experiments are conducted to validate the proposed WESNR in comparison with state-of-the-art mixed noise removal methods.

The rest of the paper is organized as follows. Section II presents in detail the proposed WESNR scheme. Section III presents the experimental results and discussions. Section IV concludes the paper.

II. WEIGHTED ENCODING WITH SPARSE NONLOCAL REGULARIZATION

A. The mixed noise

Denote by \mathbf{x} an image and by $x_{i,j}$ its pixel at location (i, j) . Let \mathbf{y} be the noisy observation of \mathbf{x} . For additive white Gaussian noise (AWGN), each noisy pixel $y_{i,j}$ in \mathbf{y} is modeled as $y_{i,j} = x_{i,j} + v_{i,j}$, where $v_{i,j}$ is i.i.d. noise and follows zero-mean Gaussian distribution. For impulse noise (IN), the two most common types of it are salt-and-pepper impulse noise (SPIN) and random-valued impulse noise (RVIN). Denote by $[d_{min}, d_{max}]$ the dynamic range of \mathbf{y} . The SPIN noise model can be described as follows: $y_{i,j} = d_{min}$ with probability $s/2$, $y_{i,j} = d_{max}$ with probability $s/2$, and $y_{i,j} = x_{i,j}$ with probability $1 - s$, where $0 \leq s \leq 1$. The RVIN noise model can be defined as: $y_{i,j} = d_{i,j}$ with probability r , and $y_{i,j} = x_{i,j}$ with probability $1 - r$, where $0 \leq r \leq 1$ and $d_{i,j}$ is uniformly distributed within $[d_{min}, d_{max}]$.

In this paper, we consider two types of mixed noise: 1) AWGN mixed with SPIN, and 2) AWGN mixed with RVIN and SPIN. For the first case, the signal observation model can be described as

$$y_{i,j} = \begin{cases} d_{min}, & \text{with probability } s/2 \\ d_{max}, & \text{with probability } s/2 \\ x_{i,j} + v_{i,j}, & \text{with probability } 1 - s \end{cases} \quad (1)$$

For the second case, the observation model is

$$y_{i,j} = \begin{cases} d_{min}, & \text{with probability } s/2 \\ d_{max}, & \text{with probability } s/2 \\ d_{i,j}, & \text{with probability } r(1 - s) \\ x_{i,j} + v_{i,j}, & \text{with probability } (1 - r)(1 - s) \end{cases} \quad (2)$$

Examples of the images corrupted by the above two types of mixed noise can be found in Figs. 7 and 8.

B. The denoising model

Many mixed noise removal methods [28]-[34] follow a two-phase framework. First, the IN pixels are detected and replaced, and then some AWGN removal methods are applied to estimate the image. The trilateral filter (TF) [27] integrates rank-order absolute difference (ROAD) statistics into BF [12] to form a simple model that does not need impulse pixel detection, and it achieves very good results for mixed AWGN and RVIN removal. However, it does not work well for either SPIN removal or mixed AWGN and SPIN removal. Furthermore, ROAD could produce false values when half of the pixels in the processing window are corrupted.

One natural question is that can we develop a mixed noise removal method which does not perform impulse pixel detection and AWGN removal separately but conducts the two tasks in a unified framework? Inspired by the robust estimation theory [41,44] and the sparse coding based image restoration techniques [17], in this paper we propose a novel weighted encoding model to remove mixed noise, which does not have an explicit impulse pixel detection step and can process AWGN and IN simultaneously. The sparsity and nonlocal self-similarity priors of natural images are also integrated into the proposed model to make it powerful for mixed noise removal.

Denote by $\mathbf{x} \in R^N$ an image. Following the notation in [13], we let $\mathbf{x}_i = \mathbf{R}_i \mathbf{x} \in R^n$ be the stretched vector of an image patch of size $\sqrt{n} \times \sqrt{n}$, where \mathbf{R}_i is the matrix operator extracting patch \mathbf{x}_i from \mathbf{x} at location i . Based on the sparse representation theory [37], we can find an over-complete dictionary $\Phi = [\phi_1; \phi_2; \dots; \phi_n] \in R^{n \times m}$ to sparsely code \mathbf{x}_i , where $\phi_j \in R^n$ is the j^{th} atom of Φ . The representation of \mathbf{x}_i over dictionary Φ can be written as $\mathbf{x}_i = \Phi \alpha_i$, where α_i is a sparse coding vector with only a few non-zero entries. The least square solution of \mathbf{x} can be obtained as

$$\mathbf{x} = \left(\sum_i \mathbf{R}_i^T \mathbf{R}_i \right)^{-1} \left(\sum_i \mathbf{R}_i^T \Phi \alpha_i \right). \quad (3)$$

For the convenience of expression, we re-write the above equation as

$$\mathbf{x} = \Phi \alpha, \quad (4)$$

where α is the set of all coding vectors α_i .

In image denoising, the observation of \mathbf{x} is noise-corrupted, and we can only encode the noisy observation \mathbf{y} over the dictionary Φ to obtain the desired α . In the case of AWGN, the encoding model can be generally written as

$$\hat{\alpha} = \arg \min_{\alpha} \|\mathbf{y} - \Phi \alpha\|_2^2 + \lambda R(\alpha), \quad (5)$$

where $R(\alpha)$ is some regularization term imposed on α and λ is the regularization parameter. With certain regularization (e.g., sparsity) term [17, 39], the resolved coding vector is the maximum a posteriori (MAP) solution for AWGN noise model. For images corrupted by mixed noise, however, the distribution of noise is generally far from Gaussian and thus the l_2 -norm data fidelity term $\|\mathbf{y} - \Phi \alpha\|_2^2$ in Eq. (5) will not lead

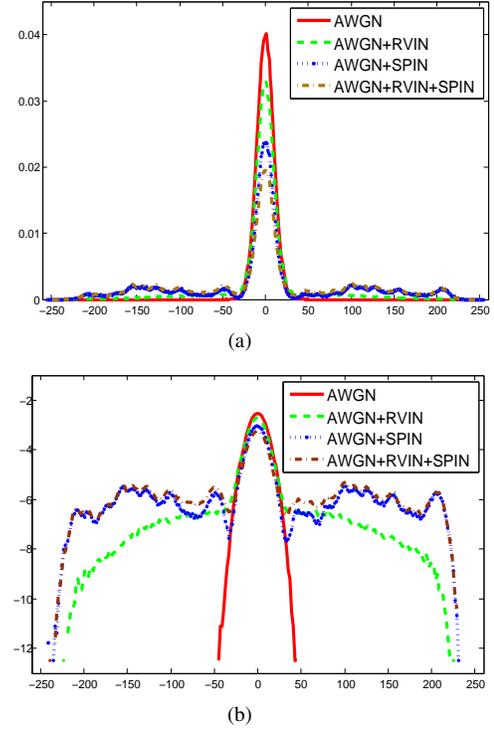


Fig. 1: The distribution of AWGN and mixed noise in (a) linear and (b) log domains, respectively.

to a MAP solution for noise removal. Let's use an example to investigate the distributions of mixed noise. We simulated AWGN (standard deviation $\sigma = 10$), RVIN ($r = 20\%$) and SPIN ($s = 40\%$), and imposed them on image Lena, respectively. Fig. 1(a) shows the distributions of data fitting residual $\mathbf{y} - \Phi \alpha$ (i.e., $\mathbf{y} - \mathbf{x}$ in the case of denoising) for AWGN, mixture of AWGN and RVIN, mixture of AWGN and SPIN, and mixture of AWGN, RVIN and SPIN, respectively. Fig. 1(b) shows these distributions in log domain to better observe the heavy tails. Compared with the Gaussian distribution, we can clearly see that the distribution of mixed noise has a heavy tail, which is caused by IN. Therefore, using the l_2 -norm to characterize the data fitting residual $\mathbf{y} - \Phi \alpha$ in Eq. (5) is not optimal in the sense of MAP estimation.

From Fig. 1, one can see that the distribution of data fitting residual is much more irregular than Gaussian, and it has a heavy tail. Intuitively, if we can modify the data fidelity term so that the residual can be more Gaussian-like, then the l_2 -norm can still be used to characterize the coding residual, making the mixed noise removal easier to handle. This motivates us to adopt the robust estimation technique [38, 41, 44] to weight the data fitting residual so that its distribution can be more regular.

Let

$$\mathbf{e} = [e_1; e_2; \dots; e_N] = \mathbf{y} - \Phi \alpha, \quad (6)$$

where $e_i = (\mathbf{y} - \Phi \alpha)(i)$. Assume that e_1, e_2, \dots, e_N are i.i.d. samples. Instead of minimizing $\|\mathbf{e}\|_2^2 = \sum_{i=1}^N e_i^2$, which actually assumes that e_i follows Gaussian distribution, we use the robust estimation technique [41, 44] to minimize the following loss:

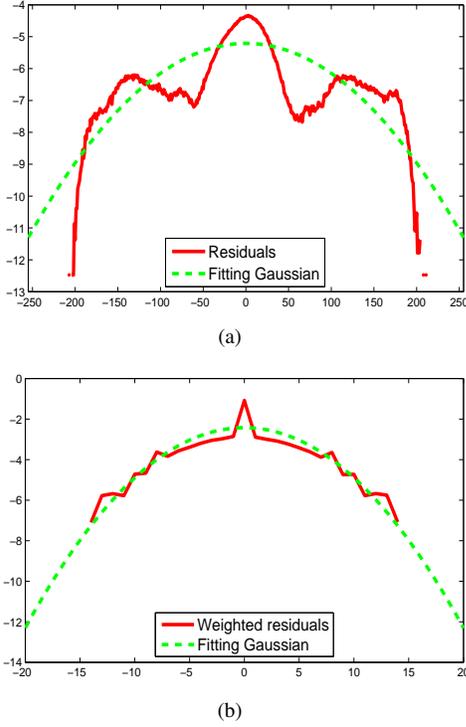


Fig. 2: The distribution of residuals e_i , weighted residuals $w_i^{1/2}e_i$ and the fitting Gaussian in log domain, respectively.

$$\min \sum_{i=1}^N f(e_i). \quad (7)$$

The function f controls the contribution of each residual to the whole loss. In general, f should have the following properties: nonnegative, monotonic, and symmetric. That is: 1) $f(e) \geq 0$ and $f(0) = 0$; 2) $f(e_i) \geq f(e_j)$ if $|e_i| \geq |e_j|$; 3) $f(e) = f(-e)$.

Obviously, when $f(e_j) = e_j^2$, the model in Eq. (7) reduces to Eq. (5). In order to weaken the effect of the heavy tail in mixed noise distribution, we can assign each residual a proper weight, resulting in a weighted residual:

$$e_i^w = w_i^{1/2}e_i. \quad (8)$$

In the problem of mixed AWGN and IN removal, the residuals can be classified into two categories. Those residuals obtained at the pixels corrupted by AWGN will basically follow Gaussian distribution and they can remain unchanged; that is, they should be assigned with weights close to 1. The residuals obtained at other pixels are mainly caused by IN, and they should be assigned with smaller weights to reduce the heavy tail of the distribution.

Let's use an example to illustrate the effect of weighting. Suppose that an image is corrupted by AWGN ($\sigma = 10$) and SPIN ($s = 40\%$). Fig. 2(a) shows the distribution of residuals e_i and the fitting Gaussian function based on the variance of e_i , Fig. 2(b) shows the distribution of weighted residuals $w_i^{1/2}e_i$ and the fitting Gaussian function based on the variance of $w_i^{1/2}e_i$ (how to set the weights will be discussed in the later development). Clearly, the distribution of weighted residuals is much closer to Gaussian distribution, implying that l_2 -norm

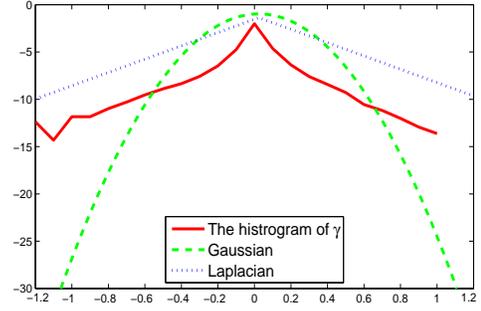


Fig. 3: The histogram of γ and the fitting Gaussian and Laplacian distributions in log domain.

can be used to model the weighted residuals for a MAP-like solution of coding vector α .

According to the above analysis, we adopt a new loss function $f(e_i) = (w_i^{1/2}e_i)^2$, and consequently we have a new model for mixed noise removal:

$$\hat{\alpha} = \arg \min_{\alpha} \|\mathbf{W}^{1/2}(\mathbf{y} - \Phi\alpha)\|_2^2 + \lambda R(\alpha), \quad (9)$$

where \mathbf{W} is a diagonal weight matrix with diagonal element $W_{ii} = w_i$. To make the above weighted encoding model more effective for mixed noise removal, some regularization terms $R(\alpha)$ can be used based on the priors of natural images. Two priors are widely used in image denoising: local sparsity and nonlocal self-similarity (NSS). The local sparsity of encoding coefficients α can be characterized by the l_1 -norm of α , while the NSS can be characterized by the prediction error of a patch by its similar patches. Inspired by the work in [17], we integrated the two priors into a sparse nonlocal regularization term and adopt it to Eq. (9).

For each patch \mathbf{x}_i , we search the similar patches to it within a large enough window centered at location i . A patch \mathbf{x}_i^q is collected as a similar patch to \mathbf{x}_i if the Euclidean distance between them is not greater than a preset threshold. Then we can select the first L closest patches to \mathbf{x}_i and use the weighted average of them, $\hat{\mathbf{x}}_i = \sum_{q=1}^L b_i^q \mathbf{x}_i^q$, to predict \mathbf{x}_i . The weight b_i^q is inversely proportional to the distance between patches \mathbf{x}_i and \mathbf{x}_i^q : $b_i^q = \exp(-\|\mathbf{x}_i - \mathbf{x}_i^q\|_2^2/h)/\omega$, where h is a preset scalar and ω is a normalization factor. If a patch and its nonlocal prediction are encoded by a given dictionary Φ_i , i. e., $\mathbf{x}_i = \Phi_i \alpha_i$ and $\hat{\mathbf{x}}_i = \Phi_i \mu_i$, then the coding coefficients α_i and μ_i should also be similar. Therefore, we can use $\sum_i \|\alpha_i - \mu_i\|_{l_p}$ as the regularization term to regularize the solution of Eq. (9):

$$\hat{\alpha} = \arg \min_{\alpha} \|\mathbf{W}^{1/2}(\mathbf{y} - \Phi\alpha)\|_2^2 + \lambda \sum_i \|\alpha_i - \mu_i\|_{l_p}, \quad (10)$$

where l_p ($p = 1$ or 2) refers to the l_p -norm.

In order to determine the value of p , we need to check the distribution of $\alpha_i - \mu_i$. Let

$$\gamma_i = \alpha_i - \mu_i. \quad (11)$$

We assume that the elements in γ_i are i.i.d. and follow generalized Gaussian distribution (GGD), which is defined as:

$$f(\gamma) = \beta \exp\{-(|\gamma|/\sigma_\gamma)^\beta\} / (2\sigma_\gamma \Gamma(1/\beta)), \quad (12)$$

where Γ denotes the gamma function, and σ_γ is a scale



Fig. 4: Five high-quality images.

parameter. The value of β in Eq. (12) determines the shape of a GGD. In particular, setting $\beta = 1$ or $\beta = 2$ leads to Laplacian distribution or Gaussian distribution, respectively.

We use an example to figure out which setting we should use for the distribution $f(\gamma)$. We run our algorithm without the regularization $\sum_i \|\alpha_i - \mu_i\|_{l_p}$ in Eq. (10) on image Lena (the algorithm is summarized in **Algorithm 1**). In Fig. 3 we plot the histogram of γ as well as the fitting Gaussian and Laplacian distributions of it. Clearly, Laplacian distribution fits the histogram of γ much better. Therefore, we approximately assume that γ follows Laplacian distribution, and hence the l_1 -norm regularization on γ could lead to a MAP-like estimation. Finally, the proposed model becomes

$$\hat{\alpha} = \arg \min_{\alpha} \{ \|\mathbf{W}^{1/2}(\mathbf{y} - \Phi\alpha)\|_2^2 + \lambda \|\alpha - \mu\|_1 \}. \quad (13)$$

In the above model, the data fidelity term weights the encoding residual, while the regularization term integrates sparsity and NSS priors. We call the proposed model weighted encoding with sparse nonlocal regularization (WESNR).

In the WESNR model Eq. (13), \mathbf{W} is a diagonal weight matrix, and its element W_{ii} is to be automatically determined and assigned to pixel i . Clearly, the pixels corrupted by IN should have small weights to reduce their effect on the encoding of \mathbf{y} over Φ , while the weights assigned to uncorrupted pixels should be close to 1. In our algorithm, the dictionary Φ is pre-learned from clean natural images (please refer to next sub-section for more information), and the pixels corrupted by IN will have big coding residuals. Therefore, the coding residual e_i can be used to guide the setting of weight W_{ii} , and W_{ii} should be inversely proportional to the strength of e_i . In order to make the weighted encoding stable and easy to control, we set $W_{ii} \in [0, 1]$. One simple and appropriate choice of W_{ii} is

$$W_{ii} = \exp(-ae_i^2), \quad (14)$$

where a is a positive constant to control the decreasing rate of W_{ii} w.r.t. e_i . With Eq. (14), the pixels corrupted by IN will be adaptively assigned with lower weights to reduce their impact in the process of encoding. Note that such a weighting scheme will make the corresponding loss function $f(e_i)$ meet the requirements 1), 2) and 3).

Once \mathbf{W} is given, the WESNR model in Eq. (13) becomes an l_1 -norm sparse coding problem and many existing l_1 -norm minimization techniques [40, 42, 43] can be used to solve it. In this paper, we solve it via the iteratively reweighted scheme [40] for its simplicity. Let \mathbf{V} be a diagonal matrix. We first initialize it as an identity matrix, and then in the $(k+1)^{th}$ iteration, each element of \mathbf{V} is updated as

$$V_{ii}^{(k+1)} = \lambda / ((\alpha_i^{(k)} - \mu_i)^2 + \varepsilon^2)^{1/2}, \quad (15)$$

where ε is a scalar and $\alpha_i^{(k)}$ is the i^{th} element of coding vector α in the k^{th} iteration. Then we update α as

$$\hat{\alpha}^{(k+1)} = (\Phi^T \mathbf{W} \Phi + \mathbf{V}^{(k+1)})^{-1} (\Phi^T \mathbf{W} \mathbf{y} - \Phi^T \mathbf{W} \Phi \mu) + \mu. \quad (16)$$

By iteratively updating \mathbf{V} and α , the desired α can be efficiently obtained. The convergence of the iteratively reweighted scheme has been proved in [40].

C. The dictionary

In sub-section II-B, we assumed that the dictionary Φ is given. The selection of dictionary is an important issue to the sparse coding and reconstruction of a signal. In particular, learning dictionaries from natural image patches has shown promising results in image restoration [13, 39]. The seminal work of K-SVD [13] learns an over-complete and universal dictionary to process any input patch; however, it is not adaptive to the content of the given patch and is not efficient due to the large number of atoms in the over-complete dictionary. In this paper, we adopt the strategy in [39] to learn a set of local PCA dictionaries from natural images.

We use the same 5 high-quality images (which are independent of the test images used in this paper) as in [39] to train the PCA dictionaries. The scenes of the five images are shown in Fig. 4. A number of 876,359 patches (size: 7×7) are extracted from the five images and they are clustered into 200 clusters by using the K-means clustering algorithm. For each cluster, a compact local PCA dictionary is learned. Meanwhile, the centroid of each cluster is calculated. For a given image patch, the Euclidian distance between it and the centroid of each cluster is computed, and the PCA dictionary associated with its closest cluster is chosen to encode the given patch. Note that since the selected dictionary, denoted by Φ_i , is orthogonal, the μ_i for patch \mathbf{x}_i can be simply computed as $\mu_i = \Phi_i^T \hat{\mathbf{x}}_i$. In our PC (3.2 GHZ CPU, 8 GB RAM memory) and under the Matlab R2011b programming environment, the patch clustering and dictionary learning process takes about 745 seconds in total. In addition, the final denoising results are not sensitive to the training images used for PCA dictionary learning. By using another five high-quality images with sufficient texture/edge regions, similar denoising results will be obtained. This is identical to the observation in [39].

D. Algorithm of WESNR

Once the dictionary Φ is adaptively determined for a given patch, the proposed WESNR model can be solved by iteratively updating \mathbf{W} and α . The updating of \mathbf{W} depends on the

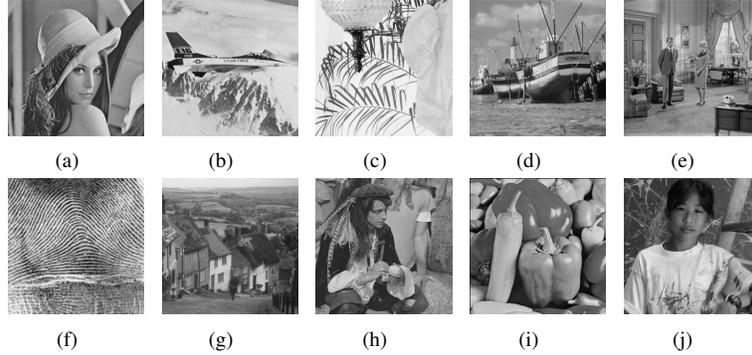


Fig. 6: The ten test images. From left to right and top to bottom: Lena, F16, Leaves, Boat, Couple, Fingerprint, Hill, Man, Peppers and Painting.

coding residual e . In the literature of mixed AWGN and SPIN noise removal [31]-[34], AMF [6] is widely used to detect SPIN. In order to make a fair comparison with them, in the case of AWGN+SPIN noise removal, we apply AMF to y to obtain an initialized image $x^{(0)}$, and then initialize e as:

$$e^{(0)} = y - x^{(0)}, \quad (17)$$

In the case of AWGN+RVIN+SPIN noise removal, AMF cannot be applied to y to initialize x . We initialize e as

$$e^{(0)} = y - \mu_y \cdot \mathbf{1}, \quad (18)$$

where μ_y is the mean value of all pixels in y and $\mathbf{1}$ is a column vector whose elements are all 1. In other words, we simply use the mean value of y to initialize x . Then the initial coding residual can be roughly computed. This simple initialization strategy works very well in all our experiments.

With the initialized coding residual $e^{(0)}$, W can be initialized by Eq. (14). The main procedures of the proposed WESNR based mixed noise removal algorithm are summarized in **Algorithm 1**.

In our algorithm, we set $t = \|\Phi\alpha^{(k+1)} - \Phi\alpha^{(k)}\|_2 / \|\Phi\alpha^{(k)}\|_2 < \tau$ as the termination condition. Fig. 5 shows the curve of $\log(t)$ versus the number of iterations by applying the WESNR algorithm to a noisy Lena image. Because of the weighting matrix W , the IN pixels in the image can be well identified and their effect is suppressed in the encoding of y . As a result, both IN and AWGN will be gradually removed in

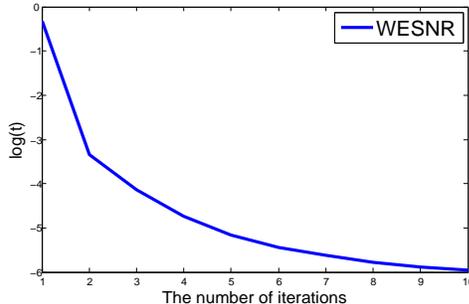


Fig. 5: A typical curve of convergence of the proposed WESNR algorithm.

Algorithm1: Mixed noise removal by WESNR

Input: Dictionary Φ , noisy image y ;
Initialize e by Eq. (17) (or Eq. (18)) and then initialize W by Eq. (14);
Initialize μ to $\mathbf{0}$.

Output: Denoised image x .

Loop: iterate on $k = 1, 2, \dots, K$;

1. Compute $\alpha^{(k)}$ by Eq. (16);

2. Compute $x^{(k)} = \Phi\alpha^{(k)}$ and update the nonlocal coding vector μ ;

3. Compute the residual $e^{(k)} = y - x^{(k)}$;

4. Calculate the weights W by $e^{(k)}$ using Eq. (14);

End

Output the denoised image $x = \Phi\alpha^{(K)}$.

the iteration. Generally, our algorithm will terminate in six to twelve iterations.

III. EXPERIMENTAL RESULTS

In this section, experiments are carried out to demonstrate the performance of the proposed WESNR algorithm. We first discuss the parameter setting in Section III-A; in Section III-B we conduct experiments on ten commonly used test images: Lena, F16, Leaves, Boat, Couple, Fingerprint, Hill, Man, Peppers and Painting, respectively (please refer to Fig. 6 for the scenes of the ten images); finally, we make some discussions in Section III-C.

A. Parameter setting

There are several parameters to set in the proposed WESNR algorithm, and they all can be easily fixed by experience. The parameter τ is to control the termination of iteration. In order to balance the denoising results and the number of iterations, we empirically set it to 0.003. The parameter a in Eq. (14) controls the decreasing rate of weights w.r.t. e and we empirically set it to 0.0008.

In Eq. (16), there are two parameters to compute the diagonal matrix V : λ and ε . In our method, the sparse nonlocal regularization is mainly to remove AWGN. In the first loop of our algorithm, since the IN is severe, the block-matching based nonlocal similar patch searching process is not accurate. Thus, the nonlocal regularization is not very helpful and we assign λ

TABLE I: PSNR (dB) and FSIM (%) results of mixed noise removal (AWGN + SPIN).

Image	$s = 30\%$				$s = 40\%$				$s = 50\%$				
	ROR-NLM	Cai et al.	l_1-l_0	WESNR	ROR-NLM	Cai et al.	l_1-l_0	WESNR	ROR-NLM	Cai et al.	l_1-l_0	WESNR	
Lena	$\sigma = 10$	29.23±0.05	31.76±0.04	33.52±0.06	33.22±0.05	28.52±0.09	31.01±0.05	32.59±0.06	32.72±0.07	24.73±0.20	30.14±0.06	31.60±0.04	31.93±0.06
		94.94±0.07	96.29±0.04	97.17±0.05	97.06±0.05	93.98±0.08	95.60±0.05	96.65±0.07	96.70±0.05	89.57±0.28	94.58±0.07	95.94±0.04	96.18±0.07
	$\sigma = 20$	26.26±0.03	29.34±0.03	30.76±0.08	30.73±0.05	25.41±0.05	28.96±0.04	30.09±0.06	30.27±0.06	22.38±0.17	28.42±0.05	29.27±0.07	29.68±0.08
		87.21±0.14	93.15±0.08	94.30±0.10	94.19±0.09	84.69±0.12	92.66±0.10	93.62±0.09	93.75±0.14	78.74±0.26	91.81±0.11	92.88±0.10	93.06±0.13
	$\sigma = 25$	25.15±0.09	28.33±0.03	29.78±0.07	29.85±0.08	24.12±0.10	28.04±0.04	29.25±0.07	29.42±0.06	21.23±0.06	27.56±0.05	28.45±0.06	28.79±0.08
		83.17±0.18	91.32±0.12	92.99±0.10	93.06±0.11	80.12±0.26	90.95±0.13	92.33±0.08	92.41±0.13	73.49±0.15	90.18±0.13	91.33±0.09	91.51±0.15
Fl6	$\sigma = 10$	28.10±0.06	30.93±0.07	32.81±0.06	32.17±0.05	27.09±0.14	29.76±0.08	31.59±0.07	31.48±0.06	24.15±0.12	28.50±0.08	30.05±0.07	30.26±0.07
		94.30±0.06	95.77±0.06	96.89±0.04	96.37±0.05	92.77±0.14	94.69±0.08	96.13±0.05	95.86±0.06	87.88±0.18	93.12±0.09	94.93±0.07	95.01±0.07
	$\sigma = 20$	25.38±0.03	28.56±0.05	29.63±0.07	29.67±0.07	24.24±0.08	27.87±0.06	28.78±0.08	28.91±0.08	21.72±0.09	26.96±0.06	27.72±0.08	27.99±0.10
		86.13±0.15	92.10±0.13	93.13±0.09	93.17±0.10	82.97±0.12	91.04±0.15	92.29±0.12	92.36±0.12	77.22±0.24	89.34±0.16	90.88±0.15	91.04±0.12
	$\sigma = 25$	24.25±0.06	27.39±0.04	28.55±0.10	28.59±0.09	22.85±0.04	26.77±0.05	27.67±0.09	27.86±0.10	20.58±0.11	25.91±0.05	26.71±0.07	26.96±0.11
		82.01±0.16	89.70±0.14	91.56±0.13	91.68±0.14	78.40±0.08	88.47±0.18	90.46±0.14	90.54±0.17	73.17±0.28	86.56±0.18	89.11±0.15	89.28±0.19
Leaves	$\sigma = 10$	24.81±0.10	28.44±0.06	30.68±0.07	30.80±0.09	23.03±0.06	26.95±0.06	28.95±0.07	29.33±0.16	20.48±0.12	25.33±0.08	27.08±0.06	27.60±0.17
		92.24±0.13	95.67±0.07	96.94±0.05	97.21±0.06	89.86±0.08	94.25±0.09	95.77±0.08	96.43±0.08	85.22±0.28	92.06±0.13	94.04±0.09	95.23±0.12
	$\sigma = 20$	23.31±0.06	26.10±0.05	27.81±0.07	27.96±0.09	21.82±0.08	25.03±0.05	26.49±0.07	26.71±0.12	19.50±0.06	23.69±0.07	24.92±0.08	25.10±0.14
		87.05±0.16	91.43±0.10	94.63±0.09	94.70±0.11	84.68±0.15	90.14±0.12	93.57±0.09	93.71±0.12	80.34±0.13	88.30±0.15	91.72±0.12	92.14±0.16
	$\sigma = 25$	22.60±0.06	25.09±0.04	26.61±0.09	26.66±0.10	21.18±0.11	24.13±0.05	25.32±0.07	25.40±0.13	18.91±0.17	22.88±0.06	23.87±0.10	23.93±0.17
		84.66±0.17	89.72±0.11	93.23±0.10	93.35±0.13	82.65±0.21	88.69±0.14	92.09±0.10	92.20±0.13	78.23±0.32	87.17±0.18	90.47±0.12	90.60±0.18
Boat	$\sigma = 10$	26.94±0.06	28.79±0.04	31.43±0.06	30.60±0.06	26.07±0.06	27.91±0.04	30.33±0.07	29.86±0.07	22.81±0.22	26.90±0.06	29.03±0.05	28.92±0.10
		93.81±0.07	94.62±0.06	96.85±0.05	96.18±0.06	92.48±0.09	93.16±0.07	95.94±0.06	95.54±0.07	87.51±0.36	91.04±0.11	94.63±0.04	94.57±0.08
	$\sigma = 20$	24.77±0.04	27.09±0.03	28.71±0.04	28.19±0.05	23.96±0.07	26.50±0.04	27.74±0.04	27.61±0.07	21.18±0.11	25.77±0.05	26.76±0.05	26.85±0.08
		87.67±0.09	91.28±0.11	93.21±0.11	92.50±0.11	85.61±0.15	89.88±0.11	91.71±0.08	91.57±0.13	80.30±0.24	87.85±0.14	90.28±0.11	90.38±0.19
	$\sigma = 25$	23.91±0.03	26.32±0.03	27.29±0.05	27.30±0.06	23.01±0.10	25.84±0.04	26.68±0.04	26.74±0.07	20.26±0.04	25.16±0.05	25.96±0.05	26.03±0.12
		84.80±0.09	89.56±0.11	90.74±0.12	90.85±0.12	82.28±0.07	88.29±0.14	90.03±0.12	90.11±0.14	76.50±0.37	86.23±0.14	88.57±0.15	88.66±0.18
Couple	$\sigma = 10$	26.82±0.04	28.53±0.03	31.12±0.04	30.55±0.07	25.93±0.09	27.66±0.04	29.92±0.06	29.84±0.05	22.81±0.22	26.73±0.04	28.81±0.06	28.95±0.06
		94.00±0.08	94.37±0.08	96.71±0.06	96.26±0.04	92.68±0.15	92.84±0.10	95.77±0.06	95.61±0.06	87.51±0.27	90.63±0.12	94.38±0.08	94.60±0.09
	$\sigma = 20$	24.70±0.03	26.81±0.02	28.36±0.05	28.08±0.04	23.90±0.05	26.25±0.02	27.51±0.06	27.55±0.05	21.18±0.06	25.57±0.03	26.64±0.08	26.85±0.05
		88.87±0.10	91.28±0.08	93.14±0.09	92.52±0.12	86.99±0.07	89.73±0.08	91.42±0.08	91.56±0.14	81.75±0.24	87.53±0.14	90.12±0.09	90.23±0.13
	$\sigma = 25$	23.84±0.03	26.07±0.03	27.12±0.07	27.18±0.05	22.94±0.05	25.60±0.03	26.50±0.07	26.68±0.04	20.41±0.12	25.02±0.03	25.89±0.05	26.03±0.06
		86.28±0.12	89.76±0.09	90.42±0.09	90.53±0.15	83.92±0.10	88.27±0.14	89.24±0.15	89.49±0.23	78.63±0.24	86.15±0.17	88.24±0.13	88.41±0.15
Fingerprint	$\sigma = 10$	26.21±0.03	27.44±0.03	29.26±0.03	28.91±0.02	24.72±0.10	26.13±0.03	27.75±0.03	28.07±0.03	21.06±0.15	24.43±0.04	26.12±0.03	27.04±0.03
		97.56±0.05	97.71±0.04	98.36±0.03	98.30±0.03	96.44±0.05	96.52±0.06	97.47±0.04	97.81±0.04	93.02±0.16	94.18±0.09	96.01±0.05	97.06±0.04
	$\sigma = 20$	23.55±0.03	24.66±0.03	26.09±0.05	26.17±0.05	22.44±0.05	23.82±0.03	25.23±0.02	25.63±0.06	19.52±0.12	22.60±0.03	24.08±0.03	24.93±0.07
		94.98±0.07	95.42±0.06	96.41±0.04	96.47±0.05	93.67±0.12	93.81±0.08	95.42±0.07	95.80±0.07	90.12±0.17	90.83±0.10	93.87±0.08	95.01±0.12
	$\sigma = 25$	22.61±0.03	23.67±0.03	25.16±0.05	25.21±0.03	21.48±0.05	22.92±0.03	24.35±0.03	24.72±0.03	18.63±0.02	21.85±0.03	23.25±0.04	24.08±0.04
		93.60±0.07	94.24±0.07	95.32±0.06	95.37±0.05	92.30±0.06	92.46±0.10	94.34±0.07	94.74±0.08	88.58±0.02	89.30±0.13	92.58±0.10	93.96±0.09
Hill	$\sigma = 10$	28.09±0.03	29.78±0.03	31.72±0.03	31.01±0.02	27.46±0.06	29.14±0.02	30.85±0.05	30.51±0.03	24.30±0.22	28.42±0.04	29.99±0.03	29.87±0.04
		93.85±0.08	94.74±0.06	96.56±0.03	95.95±0.05	92.45±0.11	93.58±0.05	95.74±0.05	95.39±0.07	88.01±0.32	91.93±0.09	94.63±0.08	94.57±0.06
	$\sigma = 20$	25.49±0.03	27.89±0.02	28.81±0.07	28.71±0.05	24.78±0.04	27.50±0.03	28.27±0.06	28.33±0.05	22.17±0.09	26.98±0.03	27.69±0.05	27.83±0.04
		86.70±0.12	91.01±0.09	92.46±0.09	92.26±0.09	84.44±0.16	89.72±0.12	91.33±0.08	91.38±0.10	79.30±0.27	87.92±0.13	90.13±0.10	90.30±0.11
	$\sigma = 25$	24.50±0.03	27.08±0.03	27.91±0.06	27.93±0.04	23.57±0.07	26.73±0.03	27.48±0.05	27.55±0.06	20.98±0.07	26.23±0.03	26.93±0.07	27.06±0.05
		83.33±0.25	89.04±0.13	90.39±0.12	90.44±0.15	80.36±0.04	87.79±0.13	89.50±0.12	89.60±0.16	75.10±0.33	86.01±0.13	88.19±0.13	88.34±0.18
Man	$\sigma = 10$	27.85±0.05	29.73±0.03	31.45±0.04	30.81±0.03	27.08±0.04	29.01±0.04	30.49±0.03	30.22±0.04	23.88±0.19	28.19±0.04	29.43±0.04	29.48±0.04
		94.34±0.07	94.97±0.05	96.47±0.05	96.05±0.04	93.19±0.07	93.79±0.07	95.59±0.06	95.46±0.06	88.73±0.26	92.13±0.08	94.35±0.07	94.61±0.09
	$\sigma = 20$	25.36±0.04	27.83±0.03	28.45±0.06	28.48±0.05	24.50±0.08	27.38±0.03	27.93±0.04	28.03±0.04	21.79±0.12	26.82±0.03	27.32±0.06	27.48±0.04
		87.47±0.09	91.61±0.08	92.25±0.09	92.33±0.08	85.11±0.19	90.41±0.08	91.26±0.10	91.36±0.12	79.61±0.27	88.76±0.11	90.27±0.10	90.35±0.12
	$\sigma = 25$	24.34±0.03	26.99±0.03	27.59±0.06	27.65±0.07	23.37±0.09	26.60±0.03	27.05±0.04	27.23±0.06	20.71±0.06	26.09±0.04	26.45±0.06	26.72±0.05
		84.02±0.16	89.73±0.10	90.57±0.10	90.66±0.11	81.37±0.23	88.54±0.13	89.55±0.12	89.62±0.13	75.59±0.28	86.82±0.16	88.42±0.17	88.57±0.20
Peppers	$\sigma = 10$	28.94±0.08	31.53±0.10	32.71±0.06	32.54±0.11	27.37±0.11	30.73±0.14	31.83±0.08	31.95±0.11	24.92±0.18	29.79±0.11	30.79±0.11	31.09±0.13
		94.97±0.06	96.78±0.04	97.37±0.06	97.17±0.04	93.89±0.08	96.29±0.06	96.61±0.07	96.82±0.05	90.39±0.21	95.59±0.06	96.14±0.05	96.32±0.06
	$\sigma = 20$	26.20±0.05	29.17±0.06	30.35±0.06	30.42±0.09	24.94±0.06	28.66±0.07	29.78±0.06	29.88±0.12	22.68±0.10	28.06±0.09	28.88±0.11	29.13±0.13
		87.91±0.10	93.68±0.08	94.53±0.09	94.69±0.08	85.68±0.17	93.39±0.10	94.25±0.06	94.36±0.09	80.80±0.18	92.83±0.18	93.47±0.09	93.59±0.10
	$\sigma = 25$	25.13±0.03	28.17±0.05	29.54±0.08	29.61±0.09	23.80±0.05	27.77±0.06	28.73±0.06	29.02±0.11	21.22±0.14	27.17±0.06	27.95±0.08	28.21±0.13
		84.28±0.14	91.99±0.11	93.44±0.08	93.61±0.08	81.69±0.02	91.74±0.12	92.74±0.07	92.96±0.13	76.19±0.27	91.28±0.12	91.85±0.09	92.13±0.13
Painting	$\sigma = 10$	29.04±0.02	31.86±0.05	33.									

a small value (0.0001 in our algorithm) to weaken the role of nonlocal regularization term. From the second loop, the IN is largely reduced, and thus the nonlocal similar patch searching becomes more accurate. Then we assign λ a large value to remove AWGN. When the standard deviation of AWGN is higher than 10, we set $\lambda = 1$; otherwise, we set $\lambda = 0.5$ to suppress AWGN while preserving the image details as much as possible. The parameter ε is a small scalar to increase the numerical stability of computing Eq. (16). We set it to

$$\varepsilon^{(k+1)} = \min(\varepsilon^{(k)}, (\text{median}(|\alpha^{(k)} - \mu|)), \quad (19)$$

with $\varepsilon^{(0)} = 0.1$. This is to ensure that ε will decrease with the iteration and it is adaptive to the range of $|\alpha^{(k)} - \mu|$. All the parameters are fixed in all our following experiments.

B. Results

We then conduct extensive experiments to demonstrate the performance of the proposed WESNR model. We consider two types of mixed noise: AWGN+SPIN, and AWGN+RVIN+SPIN. For AWGN+SPIN mixed noise, the standard deviation of AWGN varies with $\sigma = 10, 20, 25$ and the SPIN ratio varies with $s = 30\%, 40\%, 50\%$, respectively. For AWGN+RVIN+SPIN mixed noise, the standard deviation of AWGN varies with $\sigma = 5, 10, 15$, the RVIN ratio varies with $r = 5\%, 10\%, 15\%$ and the SPIN ratio varies with $s = 30\%, 40\%, 50\%$, respectively.

For AWGN+SPIN, we compare our WESNR method with the following three state-of-the-art mixed noise removal methods: ROR-NLM [30], Cai et al. [32], and l_1-l_0 [33]. Both Cai et al. [32] and l_1-l_0 [33] cannot be applied to AWGN+RVIN+SPIN mixed noise removal. Therefore, for AWGN+RVIN+SPIN we compare WESNR with ROR-NLM [30], TF [27] and BM3D [14] coupled with median filter¹ (denoted by M+BM3D, which first applies median filtering to remove IN and then applies BM3D to remove AWGN). The size of median window is set 7×7 .

The source codes of all the competing methods were obtained from the original authors. We use the default parameter settings except for Cai et al.'s method [32]. Since Cai et al.'s method is originally designed for deblurring with IN, in the experiment we set the out-of-focus kernel to have radius 0 and set $\beta_m = [0.1, 0.3, 0.3, 0.3, 0.3]$. The original setting of β_m is $[0.00001, 0.00002, 0.00002, 0.00002, 0.00002]$, but this is not suitable for mixed noise removal.

Apart from PSNR, we also compute the recently developed image perceptual quality index, FSIM [21], to evaluate quantitatively the denoising results. For each experiment, we run the programs 50 times independently, and report the mean and standard deviation of the 50 outputs. The PSNR and FSIM results on the ten test images by the competing methods are

¹The adaptive median filter (AMF) and adaptive center-weighted median filter (ACWMF) are commonly used for SPIN and RVIN detection, respectively. However, for mixed noise such as AWGN+RVIN+SPIN, to the best of our knowledge, there is no adaptive filter which can detect SPIN and RVIN simultaneously. We tested to use AMF for SPIN detection and then use ACWMF for RVIN detection, followed by BM3D for AWGN removal, the experimental results are similar to coupling BM3D with median filter but the whole algorithm becomes much more complex. Thus we use BM3D coupled with median filter for comparison.

TABLE III: Running time (second) comparison on image Lena with different levels of AWGN+SPIN.

Lena	ROR-NLM	Cai et al.	l_1-l_0	WESNR
$\sigma = 10, s = 40\%$	275	93	214	89
$\sigma = 20, s = 40\%$	313	125	165	112
$\sigma = 25, s = 40\%$	338	142	158	137

listed in Tables I and II for the two types of mixed noise, respectively. For our WESNR algorithm, the average number of iterations is about 10.

From Table I, it can be seen that for mixed AWGN+SPIN noise removal, the proposed WESNR method could consistently achieve much higher PSNR and FSIM indices than the ROR-NLM and Cai et al.'s methods, and better PSNR and FSIM performance than the l_1-l_0 method. With the increase of the strength of either AWGN or IN, the improvement of WESNR over the l_1-l_0 method is getting higher and higher. From Table II, one can clearly see that for mixed AWGN+RVIN+SPIN removal, the proposed WESNR achieves significantly better PSNR and FSIM indices than all the competing methods.

Let's give some visual comparisons of the denoising results by different methods. Fig. 7 shows the denoising results on image Lena. Fig. 7(b) and Fig. 7(c) show the Lena images corrupted by AWGN+SPIN ($\sigma = 10, s = 50\%$) and AWGN+RVIN+SPIN ($\sigma = 15, r = 15\%, s = 30\%$). In the 2nd row of Fig. 7, the denoising results by the four mixed AWGN+SPIN noise removal methods are displayed. One can see that the proposed WESNR reconstructs much cleaner and sharper image edges and generates much less artifacts, leading to visually much more pleasant denoising results than the other competing methods. The 3rd row of Fig. 7 shows the denoising results of the four mixed AWGN+RVIN+SPIN noise removal methods. One can see that M+BM3D over-smoothes much the image details and destroys the image local structure; TF results in severe SPIN caused image distortions; ROR-NLM leads to better results than TF, but it remains many Gaussian like and impulse like noises. In Fig. 8, we show the denoising results on image Leaves. Clearly, WESNR reconstructs much better the edges of leaves than all the other competing methods. Particularly, in the case of AWGN+RVIN+SPIN, all the other three methods fail to recover the image structures, while the proposed WESNR can still faithfully reconstruct the edge and texture features.

Finally, let's compare the running time of the competing methods. All the algorithms are run under the Matlab R2011b programming environment on a PC equipped with 3.2 GHz CPU and 8 GB RAM memory. Table III lists the running time (second) of the four mixed AWGN+SPIN noise removal methods in processing the image Lena (size: 512×512) with different noise levels. It can be seen that the proposed WESNR method is much faster than the other three methods. In Table IV, we list the running time (second) of the four mixed AWGN+RVIN+SPIN noise removal methods. WESNR is much faster than ROR-NLM. It is not a surprise that TF runs faster than WESNR because it is basically a type of local nonlinear filtering method. M+BM3D runs the fastest because the BM3D algorithm is implemented by C but with a Matlab



Fig. 7: Denoising results of different methods on test image Lena. (a) Original image. (b) Image corrupted by mixed noise AWGN+SPIN ($\sigma = 10, s = 50\%$). (c) Image corrupted by mixed noise AWGN+RVIN+SPIN ($\sigma = 15, r = 15\%, s = 30\%$). Second row, from left to right: the denoising results of image (b) by ROR-NLM [30], Cai et al. [32], l_1-l_0 [33] and WESNR. Third row, from left to right: the denoising results of image (c) by ROR-NLM [30], TF [27], M+BM3D [14] and WESNR.

interface.

C. Discussions

As can be seen in sub-section III-B, the proposed WESNR algorithm shows very powerful mixed noise removal performance. It can deal with either mixed AWGN+SPIN noise or mixed AWGN+RVIN+SPIN noise, and runs faster than the state-of-the-art methods in [32, 33]. The superior denoising performance of WESNR to other competing methods comes from both its weighted encoding based data fidelity term and sparse nonlocal regularization term. The role of weighted encoding is to suppress IN and the role of sparse nonlocal regularization is to suppress AWGN. Since the goal here is to remove mixed AWGN and IN noise, both of the two terms are necessary and they should work together to remove the mixed noise. Without the weighted encoding term, the IN cannot be effectively removed; without the sparse nonlocal regularization

term, the AWGN noise will largely remain in the output image. The two terms play the same important role in mixed noise removal.

In WESNR, the weights \mathbf{W} are introduced in the data fidelity term, and they are adaptively updated in the iteration process. \mathbf{W} are with real values, and the pixels corrupted by IN will be assigned small weights to reduce their effect on the encoding of \mathbf{y} over the dictionary Φ so that clean images can be reconstructed. In our algorithm, a set of orthogonal PCA dictionaries are pre-learned from some high quality images, and one local PCA dictionary is adaptively selected to process a given image patch. In a recent work [36], a weighted dictionary learning model is developed for mixed noise removal. Though both our method and Liu et al.'s method introduce weights in the data fidelity term, they have clear differences. First of all, the method in [36] mainly focuses on weighted dictionary learning, while our method focuses on weighted encoding. In [36], the dictionary is online learned

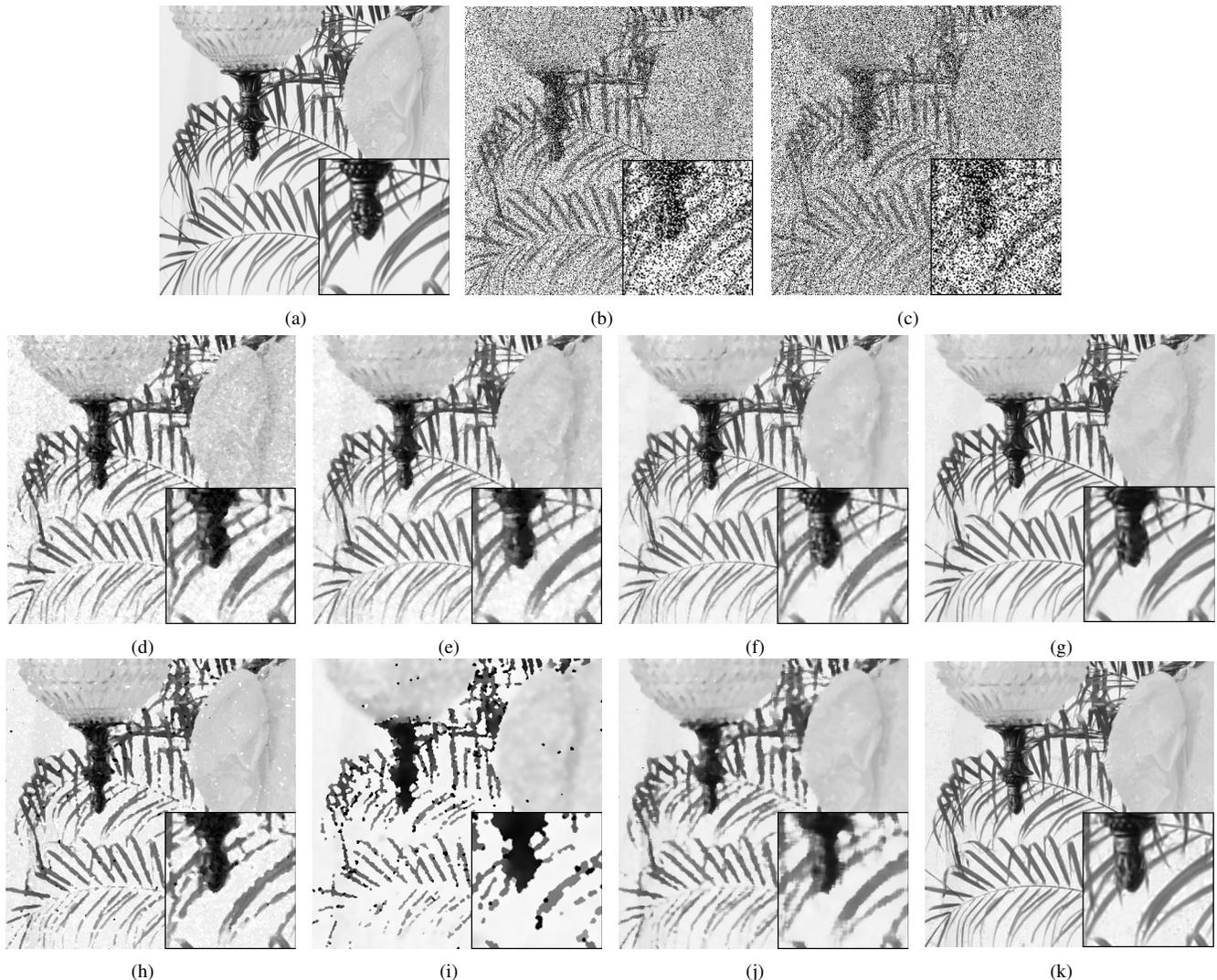


Fig. 8: Denoising results of different methods on test image Leaves. (a) Original image. (b) Image corrupted by mixed noise AWGN+SPIN ($\sigma = 20, s = 40\%$). (c) Image corrupted by mixed noise AWGN+RVIN+SPIN ($\sigma = 5, r = 5\%, s = 50\%$). Second row, from left to right: the denoising results of image (b) by ROR-NLM [30], Cai et al. [32], $l_1 - l_0$ [33] and WESNR. Third row, from left to right: the denoising results of image (c) by ROR-NLM [30], TF [27], M+BM3D [14] and WESNR.

and updated in each iteration. In our method, the dictionary is offline learned and it is fixed in the whole algorithm. That is, our algorithm is purely a sparse encoding algorithm, while the algorithm in [36] involves sparse coding and dictionary learning. The implementations of the two methods are also very different. The model in [36] is mathematically beautiful but it is somewhat complex. It needs four steps to optimize, and in each step there is a minimization problem. Our model is much simpler and it can be easily solved by iteratively re-weighted method. In another recent work [23], a 0 or 1 valued parameter is introduced in the data fidelity term and the penalty term to detect IN and remove AWGN. This is basically a detection based method, and no dictionary is used to reconstruct the image.

In the $l_1 - l_0$ algorithm [33], a dictionary is also used to reconstruct the image. However, the use of dictionary in [33] is very different from that in our method. In [33], an over-complete dictionary is online learned from the patches col-

TABLE IV: Running time (second) comparison on image Lena with different levels of AWGN+RVIN+SPIN.

	TF	ROR-NLM	M+BM3D	WESNR
$\sigma = 5, r = 5\%, s = 50\%$	9	317	4	103
$\sigma = 10, r = 10\%, s = 40\%$	9	303	4	97
$\sigma = 15, r = 15\%, s = 30\%$	9	335	4	87

lected at those outlier-free pixels in each iteration. A modified K-SVD algorithm is used for dictionary learning. The whole algorithm needs three phases to optimize. In our method, the offline learned compact PCA dictionaries are used. This is one of the reasons why our algorithm runs much faster than [33].

IV. CONCLUSION

We presented a novel model for mixed noise removal, namely weighted encoding with sparse nonlocal regularization (WESNR). The distribution of mixed noise, e.g., additive

TABLE II: PSNR (dB) and FSIM (%) results of mixed noise removal (AWGN +RVIN+ SPIN).

Image	$\sigma = 5, r = 5\%, s = 50\%$				$\sigma = 10, r = 10\%, s = 40\%$				$\sigma = 15, r = 15\%, s = 30\%$			
	TF	ROR-NLM	M+BM3D	WESNR	TF	ROR-NLM	M+BM3D	WESNR	TF	ROR-NLM	M+BM3D	WESNR
Lena	17.71±0.14	24.93±0.21	26.57±0.11	31.80±0.07	22.51±0.19	27.87±0.12	26.88±0.09	30.34±0.07	25.05±0.13	27.01±0.04	26.32±0.07	28.47±0.08
	75.51±0.50	92.23±0.27	92.04±0.18	96.67±0.06	85.72±0.27	93.10±0.09	91.58±0.09	95.32±0.06	88.65±0.17	89.81±0.12	90.77±0.08	93.11±0.13
F16	18.26±0.12	24.71±0.11	24.41±0.12	29.73±0.12	21.51±0.15	26.39±0.11	24.48±0.11	28.36±0.10	23.27±0.12	25.98±0.06	24.29±0.06	26.51±0.07
	78.76±0.34	91.50±0.30	88.70±0.13	94.86±0.10	83.05±0.23	91.67±0.13	88.03±0.11	92.93±0.14	85.83±0.17	88.71±0.14	87.37±0.13	90.31±0.15
Leaves	16.22±0.09	20.51±0.10	20.56±0.07	26.21±0.09	18.08±0.07	22.53±0.11	20.92±0.06	24.98±0.09	19.34±0.08	23.30±0.08	20.64±0.05	23.53±0.07
	72.18±0.30	86.71±0.23	84.90±0.12	92.71±0.25	77.22±0.27	88.71±0.16	84.80±0.12	89.66±0.25	80.81±0.26	87.65±0.16	83.80±0.13	88.09±0.17
Boat	16.15±0.12	22.79±0.14	23.60±0.07	28.23±0.10	20.39±0.11	25.37±0.11	23.84±0.05	27.32±0.08	22.57±0.10	25.14±0.05	23.48±0.06	26.13±0.05
	67.46±0.44	89.17±0.30	86.71±0.09	94.53±0.07	78.02±0.25	91.35±0.16	85.96±0.14	92.75±0.10	83.06±0.17	89.27±0.13	84.96±0.13	90.51±0.13
Couple	16.05±0.11	22.74±0.11	23.49±0.08	28.18±0.10	20.31±0.13	25.36±0.09	23.74±0.06	27.21±0.06	22.54±0.08	25.06±0.03	23.34±0.05	26.05±0.04
	66.51±0.40	89.01±0.22	85.66±0.16	94.53±0.13	75.65±0.20	91.66±0.09	85.05±0.09	92.96±0.07	80.26±0.14	90.05±0.10	83.96±0.13	90.77±0.14
Fingerprint	13.40±0.07	21.00±0.16	19.73±0.03	26.45±0.06	16.49±0.08	23.64±0.07	19.94±0.04	25.16±0.07	18.41±0.06	22.73±0.09	19.44±0.06	23.50±0.05
	73.65±0.29	93.10±0.24	85.35±0.11	96.57±0.06	79.24±0.28	94.38±0.08	84.50±0.16	95.04±0.06	86.66±0.21	91.01±0.12	82.63±0.22	92.43±0.07
Hill	17.25±0.11	24.56±0.18	25.67±0.12	29.66±0.06	22.03±0.16	26.90±0.06	25.95±0.07	28.66±0.05	24.34±0.10	26.13±0.04	25.41±0.05	27.31±0.06
	72.81±0.30	90.83±0.24	88.90±0.12	94.67±0.12	79.69±0.17	91.50±0.07	88.33±0.11	92.91±0.09	83.28±0.16	88.91±0.11	87.49±0.11	90.84±0.14
Man	17.02±0.10	24.02±0.16	24.99±0.13	29.10±0.06	21.44±0.18	26.47±0.10	25.25±0.10	28.13±0.06	23.75±0.10	25.91±0.05	24.81±0.06	26.80±0.04
	73.01±0.30	91.16±0.24	88.66±0.14	94.84±0.09	80.28±0.20	92.15±0.13	87.95±0.13	93.16±0.13	83.71±0.13	89.63±0.12	86.98±0.12	90.93±0.10
Peppers	18.06±0.13	25.36±0.09	26.35±0.12	29.83±0.13	22.58±0.17	27.53±0.14	26.81±0.13	28.84±0.09	24.85±0.13	26.83±0.04	26.11±0.07	27.17±0.06
	78.83±0.36	93.05±0.16	93.02±0.12	95.64±0.08	87.52±0.22	93.12±0.10	92.72±0.09	94.23±0.06	89.63±0.14	90.07±0.09	91.74±0.09	92.25±0.11
Painting	18.35±0.15	25.57±0.20	26.27±0.09	31.50±0.12	22.73±0.18	27.71±0.11	26.59±0.08	29.96±0.07	24.84±0.11	26.88±0.03	25.88±0.06	27.95±0.05
	77.13±0.32	92.77±0.17	90.34±0.07	95.78±0.08	82.88±0.21	93.06±0.10	89.80±0.06	94.27±0.06	89.66±0.12	90.49±0.07	88.69±0.09	92.19±0.07
Average	16.85±0.11	23.62±0.15	24.16±0.09	29.07±0.09	20.81±0.14	25.98±0.10	24.44±0.08	27.90±0.07	22.90±0.10	25.50±0.05	23.97±0.06	26.34±0.06
	73.59±0.36	90.95±0.24	88.43±0.12	95.08±0.10	80.93±0.23	92.07±0.11	87.87±0.11	93.32±0.10	85.16±0.17	89.56±0.12	86.84±0.12	91.14±0.12

white Gaussian noise mixed with impulse noise, is much more irregular than Gaussian noise alone, and often has a heavy tail. To address this difficulty, we adopted the weighted encoding technique to remove Gaussian noise and impulse noise jointly. We encoded the image patches over a set of PCA dictionaries learned offline, and weighted the coding residuals to suppress the heavy tail of the distribution. The weights were adaptively updated to decide whether a pixel is heavily corrupted by impulse noise or not. Meanwhile, image sparsity prior and nonlocal self-similarity prior were integrated into a single nonlocal sparse regularization term to enhance the stability of weighted encoding. The results clearly demonstrated that WESNR outperforms much other state-of-the-art mixed noise removal methods.

REFERENCES

- [1] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*. Norwell, MA: Kluwer, 1990.
- [2] D. Brownrigg, "The weighted median filter," *Commun. ACM*, vol. 27, no. 8, pp. 807-818, Aug. 1984.
- [3] S. J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Trans. Circuits syst.*, vol. 38, no. 9, pp. 984-993, Sep. 1991.
- [4] A. Nieminen, P. Heinson, and Y. Neuvo, "A new class of detail-preserving filters for image processing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 1, pp. 74-90, Jan. 1987.
- [5] T. Sun and Y. Neuvo, "Detail-preserving median based filters in image processing," *Pattern Recognit. Lett.*, vol. 15, no. 4, pp. 341-347, Apr. 1994.
- [6] H. Hwang and R. A. Haddad, "Adaptive median filters: New algorithm and results," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 499-502, Apr. 1995.
- [7] T. Chen, K. K. Ma, and L. H. Chen, "Tri-state median filter for image denoising," *IEEE Trans. Image Process.*, vol. 8, no. 12, pp. 1834-1838, Dec. 1999.
- [8] T. Chen and H. R. Wu, "Adaptive impulse detection using center-weighted median filters," *IEEE Signal Process. Lett.*, vol. 8, no. 1, pp. 1-3, Jan. 2001.
- [9] G. Pok, J. C. Liu, and A. S. Nair, "Selective removal of impulse noise based on homogeneity level information," *IEEE Trans. Image Process.*, vol. 12, no. 1, pp. 85-92, Jan. 2003.
- [10] Y. Q. Dong and S. F. Xu, "A new directional weighted median filter for removal of random-valued impulse noise," *IEEE Signal Process. Lett.*, vol. 14, no. 3, pp. 193-196, Mar. 2007.
- [11] N. I. Petrovic and V. Crnojevic, "Universal impulse noise filter based on genetic programming," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1109-1120, Jul. 2008.
- [12] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE Int. Conf. Computer Vision*, 1998, pp. 839-846.
- [13] M. Aharon, M. Elad, and A.M. Bruckstein, "K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Processing.*, vol. 54, no. 11, pp. 4311-4322, Nov. 2006.
- [14] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative Filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080-2095, Aug. 2007.
- [15] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising methods, with a new one," *Multiscale Model. Simul.*, vol. 4, no.2, pp. 490-530, 2005.
- [16] L. Zhang, W. S. Dong, D. Zhang, and G. M. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognit.*, vol. 43, no. 4, pp. 1531-1549, Apr. 2010.
- [17] W. S. Dong, L. Zhang, G. M. Shi, and X. Li, "Nonlocally Centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620-1630, Apr. 2013.
- [18] H. Burger, C. Schuler, and S. Harmeling, "Image denoising: can plain neural networks compete with BM3D," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2392-2399.
- [19] W. Dong, X. Li, L. Zhang and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 457-464.
- [20] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp.60-65.
- [21] L. Zhang, L. Zhang, X. Mou and D. Zhang, "FSIM: A Feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378- 2386, Aug. 2011.
- [22] R. Li and Y. J. Zhang, "A hybrid filter for the cancellation of mixed Gaussian noise and impulse noise," in *IEEE. Int. Conf. Information. Communications. and Signal Processing*, 2003, pp. 508-512.
- [23] M. Yan, "Restoration of images corrupted by impulse noise and mixed

- Gaussian impulse noise using blind inpainting,” *SIAM Journal on Imaging Science.*, vol. 6, no. 3, pp. 1227-1245, 2013.
- [24] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, “A new efficient approach for the removal of impulse noise from highly corrupted images,” *IEEE Trans. Image Process.*, vol. 5, no. 6, pp. 1012-1025, Jun. 1996.
- [25] S. Peng and L. Lucke, “Multi-level adaptive fuzzy filter for mixed noise removal,” in *Proc. IEEE Int. Symp. Circuits Systems*, Seattle, WA, Apr. 1995, vol. 2, pp. 1524-1527.
- [26] E. Lopez-Rubio, “Restoration of images corrupted by Gaussian and uniform impulsive noise,” *Pattern Recognit.*, vol. 43, no. 5, pp. 1835-1846, May. 2010.
- [27] R. Garnett, T. Huegerich, C. Chui, and W. He, “A universal noise removal algorithm with an impulse detector,” *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1747-1754, Nov. 2005.
- [28] L. Chih-Hsing, T. Jia-Shiuan, and C. Ching-Te, “Switching bilateral filter with a texture/noise detector for universal noise removal,” *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2307-2320, Sep. 2010.
- [29] S. Schulte, M. Nachtgaeel, V. De Witte, D. Van der Weken, and E. E. Kerre, “A fuzzy impulse noise detection and reduction method,” *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1153-1162, May. 2006.
- [30] B. Xiong and Zh. P. Yin, “A universal denoising framework with a new impulse detector and nonlocal means,” *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1663-1675, Apr. 2012.
- [31] J. F. Cai, R. Chan, and M. Nikolova, “Two-phase methods for deblurring images corrupted by impulse plus gaussian noise,” *Inverse Problem Imaging.*, vol. 2, no. 2, pp. 187-204, 2008.
- [32] J. F. Cai, R. Chan and M. Nikolova, “Fast two-phase image deblurring under impulse,” *Journal of Mathematical Image and Vision.*, vol. 36, no. 1, pp. 46-53, Jan. 2010.
- [33] Y. Xiao, T. Y. Zeng, J. Yu and M. K. Ng, “Restoration of images corrupted by mixed Gaussian-impulse noise via l_1 - l_0 minimization,” *Pattern Recognit.*, vol. 44, no. 8, pp. 1708-1720, Aug. 2010.
- [34] P. Rodríguez, R. Rojas and B. Wohlberg, “Mixed Gaussian-impulse noise image restoration via total variation,” in *IEEE. Int. Conf. Acoustics. Speech. and Signal Processing*, 2012, pp. 1077-1080.
- [35] B. Dong, H. Ji, J. Li, Z. W. Shen and Y. H. Xu, “Wavelet frame based blind image inpainting,” *Applied and Computational Harmonic Analysis*, vol. 32, no. 2, pp. 268-279, Mar. 2011.
- [36] J. Liu, X. C. Tai, H. Y. Huang, and Z. D. Huan, “A Weighted dictionary learning models for denoising images corrupted by mixed noise,” *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 1108-1120, Mar. 2013.
- [37] M. Elad, M. Figueiredo, and Y. Ma, “On the role of sparse and redundant representations in image processing,” *Proceedings of IEEE, Special Issue on Applications of Compressive Sensing and Sparse Representation.*, vol. 98, no. 6, pp. 972-982, June, 2010.
- [38] M. Yang, L. Zhang, J. Yang, and D. Zhang, “Regularized Robust Coding for Face Recognition,” *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1753-1766, May. 2013.
- [39] W. S. Dong, L. Zhang, G. M. Shi, and X. L. Wu, “Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization,” *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1838-1857, Jul. 2011.
- [40] I. Daubechies, R. Devore, M. Fornasier, and C.S. Gunturk. “Iteratively re-weighted least squares minimization for sparse recovery,” *Commun. Pure Appl. Math.*, vol. 63, no. 1, pp. 1-38, 2010.
- [41] P. J. Huber, “Robust regression: Asymptotics, conjectures and Monte Carlo,” *Ann. Stat.*, vol. 1, no. 5, pp. 799-821, 1973.
- [42] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, “A interior-point method for large-scale l_1 -regularized least squares,” *IEEE J. Sel. Topics Signal Processing.*, vol. 1, no. 4, pp. 606-617, Sep. 2007.
- [43] T. Goldstein and S. Osher, “The split Bregman method for l_1 regularized problems,” *SIAM Journal on Imaging Sciences.*, vol. 2, no. 2, pp. 323-343, 2009.
- [44] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.