Weighted Couple Sparse Representation With Classified Regularization for Impulse Noise Removal

Chun Lung Philip Chen, *Fellow, IEEE*, Licheng Liu, Long Chen, *Member, IEEE*, Yuan Yan Tang, *Fellow, IEEE*, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*— Many impulse noise (IN) reduction methods suffer from two obstacles, the improper noise detectors and imperfect filters they used. To address such issue, in this paper, a weighted couple sparse representation model is presented to remove IN. In the proposed model, the complicated relationships between the reconstructed and the noisy images are exploited to make the coding coefficients more appropriate to recover the noise-free image. Moreover, the image pixels are classified into clear, slightly corrupted, and heavily corrupted ones. Different data-fidelity regularizations are then accordingly applied to different pixels to further improve the denoising performance. In our proposed method, the dictionary is directly trained on the noisy raw data by addressing a weighted rank-one minimization problem, which can capture more features of the original data. Experimental results demonstrate that the proposed method is superior to several state-of-the-art denoising methods.**

***Index Terms*— Image denoising, couple sparse representation, dictionary learning, classified regularization, impulse noise.**

I. Introduction

IMAGE denoising is a fundamental but challenging topic and plays an important role in the image processing field [1]. Noisy images have bad characteristics, hence they can not be directly used for the subsequent image processing, e.g., segmentation, recognition, and retrieval. One of the most frequently encountered noise is the impulse noise (IN), which is mainly introduced into images by imperfect acquisition processes, transmission errors, and bit errors in analog-to-digital conversions.

The goal of image denoising is to remove noise as much as possible while preserve more image details. To suppress IN, a variety of techniques have been developed, among which the median filter (MF) [2] is widely used due to its simplicity. One limitation of MF is its poor denoising capacity since it just replaces each pixel intensity by the local median value. The improvements, such as the weighted median filter (WMF) [3] and the center weighted median (CWM) filter [4], still can not suppress IN thoroughly because they modify all pixels indifferently.

Because the IN just affects partial pixels in images and leaves the remaining ones untouched, noise detectors can be designed to identify the noisy pixels before filtering. Then the noisy pixels are filtered while the clean ones remain unchanged. Such kind of techniques are usually median- or mean-type filters based, including the adaptive center weighted median (ACWM) filter [5], Luo-iterative method [6], the contrast enhancement filter (CEF) [7], the adaptive switching median (ASWM) filter [8], the *robust outlyingness ratio* based *non-local mean* (ROR-NLM) [9], and the two-phase detector based weighted mean filter (TPD-WMF) [10]. These methods employ the noise detectors and more information from local neighborhood to estimate the center pixel. Nevertheless, they still distort some image edges and bring in many artifacts due to the limited denoising performance of the filters they used.

In [11], Garnett *et al.* presented an image statistic *rank-ordered absolute difference* (ROAD) to measure the probability of a pixel to be corrupted by IN. The ROAD is further incorporated into the bilateral filter to remove mixed noise. Later, Dong *et al.* [12] improved this idea and proposed a more robust statistic *rank-ordered logarithmic difference* (ROLD) which is integrated into the edge-preserving regularization (EPR) [12] for IN reduction.

In recent years, sparse representation (SR) [13] has been emerging as a powerful tool for handling various image processing tasks, such as denoising [14], [15], super-resolution [16]–[18], deblurring [19], inpainting [20], recognition [21]–[23], etc. The basic idea of SR is that the signal can be well reconstructed by the linear combination of a few atoms from an appropriate database called dictionary. For image denoising, image patches arranged in lexicographic order as vectors are extracted from the noisy image and served as units of SR. The reconstructed image is then obtained by averaging all the overlapping denoised patches for each position. Recent works [24]–[26] have shown that SR models have good performance in removing Gaussian noise (GN). Unfortunately, due to the totally different distributions of GN and IN, the traditional SR based methods that have fantastic performance in GN removal always fail in IN reduction.

Manuscript received September 27, 2014; revised March 9, 2015 and June 3, 2015; accepted June 29, 2015. Date of publication July 14, 2015; date of current version August 10, 2015. This work was supported in part by the Multiyear Research Grants, in part by the Macau Science and Technology Development Fund under Grant 008/2010/A1, and in part by the Macau Science and Technology Development Fund under Grant 017/2012/A1. *(Corresponding author: Licheng Liu.)*

The authors are with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: philip.chen@ieee.org; yb27408@umac.mo; longchen@umac.mo; yytang@umac.mo; yicongzhou@umac.mo).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2456432

In [27], Xiao and Zeng first incorporated noise detectors into SR model for IN removal. Later, Liu *et al.* [28] presented a weighted dictionary learning model for removing mixed noise (GN mixed with IN), in which the dictionary learning, image reconstruction, noise detection, and parameter estimation are integrated into a four-step framework. Recently, by forcing the distribution of the sparse coding residual more Gaussian-like, Jiang *et al.* [29] proposed a weighted encoding model to handle the mixed noise. Instead of employing noise detectors, they chose to detect noisy pixels by a soft rule according to the encoding residual.

Though the SR based methods mentioned above have achieved encouraging denoising performance for mixed noise, they may face some troubles when deal with the single IN removal problem. Firstly, none of those methods takes into account the IN characteristics, hence many image details may be lost when they are used to remove single IN. Secondly, since just the clean pixels are used for sparse coding, these methods may fail in the high noise level environment, where the number of useful clean pixels are limited. Finally, the values of clean pixels are also changed in the reconstruction stage, while for IN removal, leaving the clean pixels unchanged can significantly improve the quality of reconstructed image.

To address the above concerns, in this paper we proposed a Weighted Couple SR (WCSR) model for IN removal. In our proposed method, a weight matrix is incorporated to classify the pixels into three categories and to subtly determine the contribution of each pixel offered in the sparse coding stage, which makes the SR model more suitable for IN removal. Besides, the reconstructed and noisy images are coupled in sparse coding to explore the complicated relationships between them. More specifically, because the reconstructed image contains less noise and share the similar scenario with the original noise-free one, it is used in our method to compensate the lost information in the noisy image. Coupled coding of the reconstructed and noisy images yields the sparse coefficients much closer to those of the underlying degradation-free image than the coefficients that would result from coding of the single noisy image. Therefore, the sparse coding coefficients produced by the proposed WCSR model are more appropriate for reconstruction. The contributions of this paper are listed as follows.

- The cleanliness of each pixel is measured by a fuzzy membership function, which is more suitable for IN since the inherent feature of IN is uncertainty [30]. Moreover, the weight determines how much contribution each pixel should offer in the sparse coding stage, which can suppress the effects of outliers.
- The WCSR model simultaneously codes the noisy and the reconstructed images to explore the complicated relationships between them, which helps to improve the reconstruction accuracy.
- All the image pixels are classified into three categories, and different categories are assigned with different regularizations according to their characteristics. This preserves more image details and makes the proposed model more robust to outliers.
- To search the best basis to represent the noisy image, we propose a novel dictionary learning method to train the dictionary directly on the raw data with outliers. The dictionary is learned by jointly updating the dictionary atoms and the corresponding coefficient vectors, which is convergent and efficient.

The remainder of this paper is organized as follows. In Section II, the noise detection rule and the proposed weighted sparse-land model are described in detail. Experimental results are shown in Section III. Finally, section IV reaches a conclusion.

II. The Proposed Algorithm

In this section, we first introduce the IN model and the detection rule. Then we describe in detail the proposed WCSR model and its solutions. Throughout the paper, we use the uppercase and lowercase letters to denote vectors (images) and entries (pixels), respectively.

A. Impulse Noise Model

Let $x_{i,j}$ and $o_{i,j}$ denote the (i, j)-th pixel values in the corrupted and clean images, respectively. Supposing the pixel values are bounded by m_{min} and m_{max}, the IN model is described as,

$$x_{i,j} = \begin{cases} o_{i,j}, & \text{with probability } 1-p, \\ u_{i,j}, & \text{with probability } p, \end{cases} \tag{1}$$

where $u_{i,j}$ is the IN value which is independent with the original pixel value $o_{i,j}$, and p is the IN probability. Actually, the pixel $x_{i,j}$ in IN corrupted image can be viewed as a Bernoulli random variable (e.g., $x_{i,j} \sim \text{Bern}(p)$) with outcomes $\{u_{i,j}, o_{i,j}\}$. There are two kinds of IN: salt and pepper noise (SPN) and random-valued impulse noise (RVIN). For SPN, $u_{i,j}$ equals to m_{min} or m_{max}, while for RVIN, $m_{min} \leq u_{i,j} \leq m_{max}$. In 8-bit grayscale images, $m_{min} = 0$ and $m_{max} = 255$.

B. Outlier Detection

In our approach, we use two different rules for SPN and RVIN detection, namely "hard rule" for the former and "soft rule" for the latter. The reason is that SPN corrupted pixel values are extremely large or small, which can be easily detected. It is sufficient to detect them in a "hard rule". While for RVIN, the noisy pixel values may not be so different from those of corresponding clean pixels, hence it is more appropriate to detect them in a "fuzzy way" [30].

The adaptive median filter (AMF) [31] is used in our approach to detect SPN. For simplicity, suppose Y is the filtered result by a median-type filter, the detection result of AMF is recoded in a binary matrix W,

$$w_{i,j} = \begin{cases} 0; & x_{i,j} = y_{i,j} \text{ and } y_{i,j} \in \{0, 255\} \\ 1; & \text{otherwise} \end{cases} \tag{2}$$

in which $w_{i,j} = 0$ means the pixel $x_{i,j}$ be corrupted by noise, while $w_{i,j} = 1$ denotes $x_{i,j}$ be clean.

For RVIN, the "soft detection rule" is achieved by introducing a fuzzy membership function on the ROLD values of the noisy image. The ROLD is a local image statistic describing the noisy pixels based on the assumption that the absolute differences between the noisy pixels and their neighbors are relatively larger than those between the clean ones and their neighbors. Readers are referred to [11] for more details of ROLD. Here, we choose the exponential function for simplicity,

$$w_{i,j}=\begin{cases}1; & ROLD(x_{i,j})\le\tau_1\\ \exp\left(-\frac{(\alpha-ROLD(x_{i,j}))^2}{\sigma^2}\right); & \tau_1 < ROLD(x_{i,j})\le\tau_2\\ 0; & ROLD(x_{i,j})>\tau_2\end{cases} \tag{3}$$

where $0 \le w_{i,j} \le 1$ denotes the cleanliness of pixel $x_{i,j}$. Especially, $w_{i,j} = 1$ means $x_{i,j}$ to be a clean pixel and $w_{i,j} = 0$ indicates $x_{i,j}$ to be a noisy one. We set the parameter $\sigma = 1$ in our method.

In this paper, all the pixels in noisy image are divided into three categories based on the fuzzy weights, namely, clean, slightly corrupted, and heavily corrupted pixels. For simplicity, we use $\mathcal{C} = \{(i, j)|w_{i,j} = 1\}$, $\mathcal{S} = \{(i, j)|0 < w_{i,j} < 1\}$, and $\mathcal{H} = \{(i, j)|w_{i,j} = 0\}$ to denote the coordinates of these three classes, respectively. In the following, we will denote $X_{\mathcal{C}}$, $X_{\mathcal{S}}$, and $X_{\mathcal{H}}$ as the set of clean pixels, the set of slightly corrupted pixels, and the set of heavily corrupted pixels, respectively. Obviously, $X_{\mathcal{S}}$ is null for SPN corrupted images.

C. Weighted Sparse-Land Model

For an image contaminated by IN, not all the pixel values are changed, and there are still some clean pixels that are useful. One can use these clean pixels to reconstruct the original image. For sparse-land model, we should exclude the noisy pixels and restrain the sparse coding on these clean ones. This can be achieved by introducing a weight matrix into the SR model [32],

$$\{\hat{\alpha}_{i,j}\} = \arg\min_{\alpha_{i,j}} \|R_{i,j}W \otimes (R_{i,j}X - D\alpha_{i,j})\|_2^2 \quad s.t.\ \|\alpha_{i,j}\|_0 \le L \tag{4}$$

where the symbol $\otimes$ denotes Hadamard (element-wise) product; X is the corrupted image with size of $\sqrt{N} \times \sqrt{N}$. W is a weight matrix generated by a noise detector, and with the same size of X; $D \in R^{n\times K}$ $(n < K)$ is a redundant dictionary, whose each column represents an atom; $R_{i,j}$ denotes a $n \times N$ matrix that is used to extract the (i, j)-th $\sqrt{n} \times \sqrt{n}$ patch (reshaped as a vector) from the image; α is the representation coefficient expected to be sparse, that is, just a few entries in α are nonzero; L is the sparse ratio.

The above model is called weighted sparse representation (WSR). The basic assumption of SR based denoising framework is that the noisy and original images share the same coefficients on a certain dictionary. Hence once the coefficients are calculated from the noisy patches, the reconstructed patches can be estimated by the product of the dictionary and the coefficients. The restored image is then directly obtained by averaging all the overlapping reconstructed patches,

$$\hat{X} = \left(\sum_{i,j} R_{i,j}^T R_{i,j}\right)^{-1} \cdot \sum_{i,j} R_{i,j}^T D\alpha_{i,j} \tag{5}$$

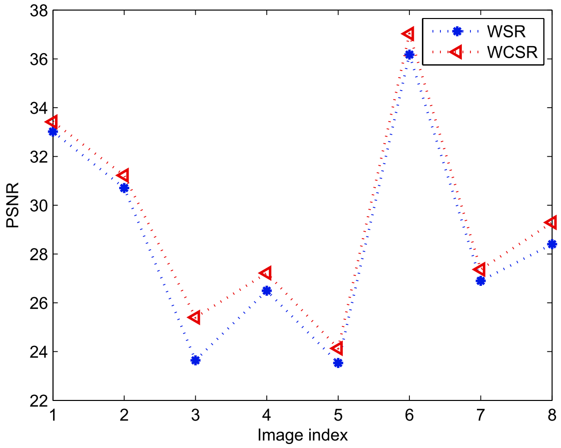


Fig. 1. PSNR values of recovered images by using WSR (Eq. (4)) and WCSR (Eq. (6)). The x-coordinate denotes the image index shown in Fig. 2: 1: *lena*, 2: *pepper*, 3: *barbara*, 4: *boat*, 5: *bridge*, 6: *house*, 7: *pentagon*, 8: *F16*.

Though WSR performs well in suppressing most of the IN, it has two main shortcomings. First, since it uses clean pixels for sparse coding, model (4) will be failed when the noise density is high, where the number of clean pixels can be used is limited. Second, the weight matrix in (4) generated by the noise detector is not always credible, especially for RVIN corrupted image. Some noisy pixels may be judged as clean ones and used for sparse coding. This will cause severe distortion of the sparse coefficients and make them not able to accurately reconstruct the noise-free image. Inspired by the fact that the reconstructed image contains less noise but the same scenario of the original noise-free one, a weighted couple SR (WCSR) model is presented to simultaneously encode the reconstructed and the noisy images. In WCSR, the reconstructed image is used to compensate the lost information in the noisy one, which drags as much as possible the sparse coefficients back to those that would generated from coding of the noise-free image. Hence the coefficients generated by the WCSR are more appropriate for image reconstruction. The WCSR model is formulated as follows,

$$\begin{aligned}\{\hat{\alpha}_{i,j}\} &= \arg\min_{\alpha_{i,j}} \left\{ \sum_{i,j} \frac{1}{2}\|R_{i,j}W \otimes (R_{i,j}X - D\alpha_{i,j})\|_2^2 \right.\\ &\qquad \left. + \sum_{i,j} \frac{1}{2}\|R_{i,j}(I - W) \otimes (R_{i,j}Y - D\alpha_{i,j})\|_2^2 \right\}\\ &\quad s.t.\ \|\alpha_{i,j}\|_0 \le L\end{aligned} \tag{6}$$

where Y denotes the estimation of X, I is a matrix with proper size, and its entries all are ones.

To verify the superiority of WCSR (Eq. (6)) over WSR (Eq. (4)), we utlize the two models to restore all the tested images (shown in Fig. 2) corrupted by RVIN with 40% noise density, respectively. The dictionary D is chosen as DCT dictionary. Fig. 1 shows the PSNR values of

Fig. 2. Test images. From left to right: *lena*, *pepper*, *barbara*, *boat*, *bridge*, *house*, *pentagon*, and *F16*.

restoration results. From this figure, we can see that, the results generated by the WCSR are better than those of the WSR for all the tested images. This owes to the fact that the information of reconstructed image utilized in the sparse coding helps to generate more appropriate coefficients to estimate the noise-free image.

Inspired by the fact that the global prior knowledge can further improve the denoising performance of SR model [14], we choose to minimize the following regularization term

$$\|W \otimes (Y - X)\|_2^2 \tag{7}$$

as the global prior. It can been see that (7) forces the estimations close to the clean pixels, that is, the pixels in the reconstructed image should approximate to these corresponding clean ones as much as possible. This preserves more image details and improves the quality of final output.

Though the l_2-norm in (7) makes the final model well in keeping the image details, it is sensitive to the noise detector that generates the weight matrix, especially when the noise level is high. To enhance its robustness, one thing we can do is to introduce a l_1-norm penalty term into the SR model, since l_1-norm has been demonstrated more suitable for IN than l_2-norm [33]. According to [33], minimizing of l_1-norm involves an implicit detection of IN which is very crucial in IN reduction.

In [27], Xiao *et al.* introduced the following l_1-term on noise candidates into SR model to make it more robust to outliers,

$$\sum_{(i,j)\in\mathcal{N}} \|y_{i,j} - x_{i,j}\|_1 \tag{8}$$

where $y_{i,j}$ denotes the (i, j)-th estimation, $\mathcal{N}$ is the index set of all the noisy pixels. Minimization of (8) means that the estimations should close to noisy pixels under the rule of l_1-norm data fidelity.

One disadvantage of (8) is that all the noisy pixels are covered in the l_1-term. Unfortunately, the heavily corrupted pixels have no relationships with the corresponding original clean ones, and their use in the loss function can only be harmful. On the contrary, for the slightly corrupted pixels, although their values are also changed, the biases are relatively smaller, and the original pixel values can still be mined by the l_1-norm. Inspired by this, we choose to use the following l_1-term,

$$\sum_{(i,j)\in\mathcal{S}} \|(1 - w_{i,j})\left(y_{i,j} - x_{i,j}\right)\|_1 \tag{9}$$

Note that the difference between (9) and (8) is that in (9) just the slightly corrupted pixels are considered in l_1-norm, and there is a weight determining how much contribution each pixel should offer.

Combining the WCSR with (7) and (9), we present the following IN removal model which is named as WCSR-l_2l_1,

$$\begin{aligned}(\hat{\alpha}, \hat{D}, \hat{Y}) &= \arg\min_{\alpha,D,Y}\Bigg\{\sum_{i,j}\frac{1}{2}\|R_{i,j}W \otimes \left(R_{i,j}X - D\alpha_{i,j}\right)\|_2^2 \\ &\quad + \sum_{i,j}\frac{1}{2}\|R_{i,j}(I - W) \otimes \left(R_{i,j}Y - D\alpha_{i,j}\right)\|_2^2 \\ &\quad + \frac{\lambda_1}{2}\|W \otimes (Y - X)\|_2^2 \\ &\quad + \lambda_2 \sum_{(i,j)\in\mathcal{S}} \|(1 - w_{i,j})(y_{i,j} - x_{i,j})\|_1\Bigg\} \\ &\quad s.t.\ \|\alpha_{i,j}\|_0 \le L\end{aligned} \tag{10}$$

where λ_1 and λ_2 are two control parameters. By introducing another matrix $\bar{W}$ with each entry $\bar{w}_{i,j} = (1 - w_{i,j}) \cdot sign(w_{i,j})$, where *sign* is the symbolic function. Model (10) is rewritten as,

$$\begin{aligned}(\hat{\alpha}, \hat{D}, \hat{Y}) &= \arg\min_{\alpha,D,Y}\Bigg\{\sum_{i,j}\frac{1}{2}\|R_{i,j}W \otimes \left(R_{i,j}X - D\alpha_{i,j}\right)\|_2^2 \\ &\quad + \sum_{i,j}\frac{1}{2}\|R_{i,j}(I - W) \otimes \left(R_{i,j}Y - D\alpha_{i,j}\right)\|_2^2 \\ &\quad + \frac{\lambda_1}{2}\|W \otimes (Y - X)\|_2^2 + \lambda_2\|\bar{W} \otimes (Y - X)\|_1\Bigg\} \\ &\quad s.t.\ \|\alpha_{i,j}\|_0 \le L.\end{aligned} \tag{11}$$

The proposed WCSR-l_2l_1 model that aims to optimize three variables simultaneously is not easy to solve. A tractable optimization problem can be obtained by relaxing (11). Fixed the other two variables, the optimization of the third one changes into a convex subproblem, which admits an efficient solution.

D. Alternating Minimization

In this subsection, we introduce an alternating minimization method to solve the proposed model. The three variables α, D, and Y can be updated alternatively by solving three optimization subproblems.

(1) Sparse coding: given Y, D, the coefficient α is calculated by

$$\begin{aligned}\left\{\hat{\alpha}_{i,j}\right\} &= \arg\min_{\alpha}\Bigg\{\sum_{i,j}\frac{1}{2}\|R_{i,j}W \otimes \left(R_{i,j}X - D\alpha_{i,j}\right)\|_2^2 \\ &\quad + \sum_{i,j}\frac{1}{2}\|R_{i,j}(I - W) \otimes \left(R_{i,j}Y - D\alpha_{i,j}\right)\|_2^2\Bigg\} \\ &\quad s.t.\ \|\alpha_{i,j}\|_0 \le L\end{aligned} \tag{12}$$

by introducing the following auxiliary variables,

$$\tilde{R}_{i,j}\tilde{W}=\begin{pmatrix} R_{i,j}W \\ R_{i,j}(I-W)\end{pmatrix},\quad \tilde{R}_{i,j}\tilde{Y}=\begin{pmatrix} R_{i,j}X \\ R_{i,j}Y\end{pmatrix},\quad \tilde{D}=\begin{pmatrix} D \\ D\end{pmatrix} \tag{13}$$

(12) is rewritten as,

$$\left\{\hat{\alpha}_{i,j}\right\} = \arg\min_{\alpha_{i,j}} \sum_{i,j} \left\| \tilde{R}_{i,j}\tilde{W} \otimes \left(\tilde{R}_{i,j}\tilde{Y} - \tilde{D}\alpha_{i,j}\right)\right\|_2^2 \quad s.t.\ \|\alpha_{i,j}\|_0 \le L \tag{14}$$

Eq. (14) is a weighted sparse coding problem due to the existence of the weight matrix and the Hadamard product operator. In [27], forcing the weight to be a binary matrix, Xiao and Zeng, *et al.* solved such a problem by using the strategy of SR based image inpainting, in which the OMP algorithm is chosen and slightly changed by projecting only the clean pixels on the dictionary. Unfortunately, that strategy no longer works for our problem because the weight $\tilde{W}$ in (14) is a fuzzy weight matrix with each entry $0 \le \tilde{w}_{i,j} \le 1$. To solve Eq. (14), we derive each $\alpha_{i,j}$ by a minimization problem as follows,

$$\hat{\alpha}_{i,j} = \arg\min_{\alpha} \left\| Q(R_{i,j}\tilde{Y}) - Q\tilde{D}\alpha_{i,j}\right\|_2^2 \quad s.t.\ \|\alpha_{i,j}\|_0 \le L \tag{15}$$

where $Q = \mathrm{diag}\big(\tilde{R}_{i,j}\tilde{W}\big)$ is a diagonal matrix. Replaced by $\bar{y}_{i,j} = Q\cdot(R_{i,j}\tilde{Y})$ and $\bar{D} = Q\cdot\tilde{D}$, the problem is transformed into,

$$\hat{\alpha}_{i,j} = \arg\min_{\alpha_{i,j}} \left\|\bar{y}_{i,j} - \bar{D}\alpha_{i,j}\right\|_2^2, \quad s.t.\ \|\alpha_{i,j}\|_0 \le L \tag{16}$$

which is a sparse coding problem of projecting the signal $\bar{y}_{i,j}$ on the new dictionary $\bar{D}$, and can be efficiently solved by OMP algorithm.

(2) Dictionary learning: given Y and α, the dictionary D is updated as

$$\hat{D} = \arg\min_{D} \sum_{i,j} \frac{1}{2}\|R_{i,j}W \otimes \left(R_{i,j}X - D\alpha_{i,j}\right)\|_2^2 + \sum_{i,j} \frac{1}{2}\|R_{i,j}(I-W) \otimes \left(R_{i,j}Y - D\alpha_{i,j}\right)\|_2^2 \tag{17}$$

Note that there are two dictionaries, one is for representing the reconstructed image Y, the other is for decomposing the noisy image X. Indeed, (17) suggests we train the two dictionaries simultaneously by combing the two dictionaries as a new one, $D^{new} = \left(D^T, D^T\right)^T$, however, this will be time consuming. Actually, we experimentally found that it works well if we manually set the two dictionaries equally and just train one on X. This is because Y is the estimation of X, then the trained dictionary $\hat{D}$ can also reduce the residual term $\sum_{i,j}\frac{1}{2}\|R_{i,j}(I-W)\otimes(R_{i,j}Y-\hat{D}\alpha_{i,j})\|_2^2$. Hence the dictionary is trained as,

$$\hat{D} = \arg\min_{D} \sum_{i,j} \|R_{i,j}W \otimes \left(R_{i,j}X - D\alpha_{i,j}\right)\|_2^2. \tag{18}$$

Due to the existence of weight matrix, the above problem can not be directly solved by the K-SVD [34] algorithm. Denote by $W_{\mathbb{B}} = [R_{i,j}W] \in \mathbb{R}^{n\times N}$, $X_{\mathbb{B}} = [R_{i,j}X] \in \mathbb{R}^{n\times N}$, and $A = [\alpha]_{i,j} \in \mathbb{R}^{K\times N}$, where $i, j \in 1, 2, \cdots, \sqrt{N}$, (18) is rewritten as,

$$\hat{D} = \arg\min_{D,\|d_k\|=1} \|W_{\mathbb{B}} \otimes (X_{\mathbb{B}} - DA)\|_2^2 \tag{19}$$

A natural choice is to update one atom d_k of D with other atoms fixed each time,

$$\left(\hat{d}_k, \hat{\alpha}_X^k\right) = \arg\min_{\alpha_X^k, \|d_k\|_2^2=1} \|W_{\mathbb{B}} \otimes \left(E^k - d_k\alpha_X^k\right)\|_2^2 \tag{20}$$

where α_X^k is a row vector composed by all the non-zero entries of the k-th row of coefficient matrix A. $E^k = X_{\mathbb{B}} - \sum_{l=1,l\neq k}^{K} d_l\alpha_X^l$ is the residual error corresponding to the training samples that currently use the d_k. This is a weighted rank one matrix approximation problem, which cannot be directly solved via KSVD but can be solved by the iterative weighted KSVD algorithm [35], [36]. However, the weighted KSVD can not work for the unweighted case when $W = \tau I$ is a scalar matrix [28].

By adopting the alternating optimization strategy [37], we introduce a dictionary learning method to efficiently approximate the solution of (20). To update d_k and α_X^k, we have the following property.

Property 1: For a multivariate function $F(d,\alpha) = \|W \otimes (E - d\alpha)\|_F^2$, the minimum point $(\hat{d}, \hat{\alpha})$ is calculated as $\hat{d} = \Phi^{-1}\cdot(\tilde{W}\otimes E\alpha^T)$, and $\hat{\alpha} = d^T(\tilde{W}\otimes E)\cdot\Psi^{-1}$, where $\tilde{W} = W \otimes W$, $\Phi = \mathrm{diag}(W(\alpha\otimes\alpha)^T)$, and $\Psi = \mathrm{diag}\big(\tilde{W}^T\cdot(d\cdot d)\big)$.

Proof: See Appendix. ■

According to Property 1, the solutions of (20) are given by

$$\begin{cases} \hat{d}_k = \Phi^{-1}\cdot\left(\left(\tilde{W}\otimes E^k\right)\cdot(\alpha_X^k)^T\right) \\ \hat{\alpha}_X^k = (d_k)^T\cdot\left(\tilde{W}\otimes E^k\right)\cdot\Psi^{-1} \end{cases} \tag{21}$$

The convergence is guaranteed and achieved by a single iteration [37].

(3) Image estimation: given D and α, the reconstructed image Y can be estimated from the following minimization problem,

$$\hat{Y} = \arg\min_{Y}\left\{\sum_{i,j}\frac{1}{2}\|R_{i,j}(I-W)\otimes\left(R_{i,j}Y - D\alpha_{i,j}\right)\|_2^2 + \frac{\lambda_1}{2}\|W\otimes(Y-X)\|_2^2 + \lambda_2\|\bar{W}\otimes(Y-X)\|_1\right\} \tag{22}$$

By introducing another variable $U = Y - X$, the above objective function is replaced by,

$$G(U) = \sum_{i,j}\frac{1}{2}\|R_{i,j}(I-W)\otimes(R_{i,j}U + R_{i,j}X - D\alpha_{i,j})\|_2^2 + \frac{\lambda_1}{2}\|W\otimes U\|_2^2 + \lambda_2\sum\|\bar{W}\otimes U\|_1 \tag{23}$$

calculating the derivative of $G(U)$ with respect to U, and considering the properties of Hadamard product, we have

$$\frac{\partial G(U)}{\partial U} = ((I-W)\otimes(I-W))\otimes(M\otimes U + M\otimes X - Z) + \lambda_1(W\otimes W)\otimes U + \lambda_2(\bar{W}\otimes\bar{W})\otimes\mathrm{sign}(U) \tag{24}$$

where $M = \sum_{i,j} R_{i,j}^T R_{i,j}$ and $Z = \sum_{i,j} R_{i,j}^T D\alpha_{i,j}$. Here both M and Z have their own physical meanings, e.g., M denotes the overlapping weight matrix, and Z is the estimation via the dictionary without averaging.

Setting the derivative to zero, (24) can be solved in an element-wise manner. Let $u_{i,j}$, $m_{i,j}$, and $z_{i,j}$ be the entry of U, M, and Z, respectively. Considering each (i, j)-th pixel, $u_{i,j}$ can be calculated according to the following three cases.

Case 1: If the (i, j)-th pixel is a clean one, that is, the weight for this pixel is $w_{i,j} = 1$, then according to (24) the solution is,

$$u_{i,j} = 0. \tag{25}$$

Case 2: If the (i, j)-th pixel is a heavily corrupted one, which means the corresponding weight is $w_{i,j} = 0$, according to (24) we have,

$$u_{i,j} = \frac{z_{i,j}}{m_{i,j}} - x_{i,j}. \tag{26}$$

Case 3: If the (i, j)-th pixel is a slightly corrupted one, and the corresponding weight is $0 < w_{i,j} < 1$. Then the solution can be obtained by the following equation,

$$\begin{aligned}\frac{\partial G(u_{i,j})}{\partial u_{i,j}} &= \left(1 - w_{i,j}\right)^2 \left(m_{i,j}u_{i,j} + m_{i,j}x_{i,j} - z_{i,j}\right)\\ &\quad + \lambda_1 w_{i,j}^2 u_{i,j} + \lambda_2 w_{i,j}\,\mathrm{sign}(u_{i,j})\\ &= 0.\end{aligned} \tag{27}$$

According to [40], (27) has a closed form formula for the optimal solution for $u_{i,j}$, which is a shrinkage operation,

$$u_{i,j} = S(f(x_{i,j}, z_{i,j}), v_{i,j}) \tag{28}$$

where S is the soft threshold operator, defined as,

$$S(t, \lambda) = \begin{cases} t - \lambda; & t > \lambda \\ 0; & -\lambda \le t \le \lambda \\ t + \lambda; & t < -\lambda \end{cases} \tag{29}$$

and $f(x_{i,j}, z_{i,j})$ and $v_{i,j}$ are as follows,

$$f(x_{i,j}, z_{i,j}) = \frac{\left(1 - w_{i,j}\right)^2 \left(z_{i,j} - m_{i,j}x_{i,j}\right)}{\left(1 - w_{i,j}\right)^2 m_{i,j} + \lambda_1 w_{i,j}^2} \tag{30}$$

$$v_{i,j} = \frac{\lambda_2 w_{i,j}}{\left(1 - w_{i,j}\right)^2 m_{i,j} + \lambda_1 w_{i,j}^2} \tag{31}$$

Note that $Y = U + X$, then the estimated value for the (i, j)-th pixel is given by,

$$y_{i,j} = \begin{cases} x_{i,j}; & \text{if } w_{i,j} = 1 \\ \frac{z_{i,j}}{m_{i,j}}; & \text{if } w_{i,j} = 0 \\ x_{i,j} + S(f(x_{i,j}, z_{i,j}), v_{i,j}); & \text{if } 0 < w_{i,j} < 1 \end{cases} \tag{32}$$

Algorithm 1 The Proposed Algorithm for IN Removal

```
Input: The noisy image $X$; Maximum number of iterations $k_{max}$, $\tau_1$, and $\tau_2$; $\lambda_1$ and $\lambda_2$ in (11);
1: Initialize $k = 0$, $Y^{(k)} = X$, $\tau_1^{(k)} = \tau_1$, $\tau_2^{(k)} = \tau_2$, the initial weight $W^{(k)}$ is generated on $X$ via (2) or (3), calculate the initial coefficient $\alpha^{(k)}$ via (4);
2: Iterative on $k$ until convergence or $k = k_{max}$
     1) Compute the auxiliary estimation $Y^{(k+\frac{1}{2})} = D\alpha^{(k)}$.
     2) Generate the weight matrix $W^{(k)}$ for $Y^{(k)}$ via (2) for SPN and (3) with $\tau_1^{(k)}$ and $\tau_2^{(k)}$ for RVIN.
     3) Update coefficient: replace $Y$ and $X$ in (12) by $Y^{(k+\frac{1}{2})}$ and $Y^{(k)}$, then calculate $\alpha^{(k)}$.
     4) Update dictionary: if mod(k,q)=0, update $D$ from $Y^{(k)}$ by (18).
     5) Compute matrices $M$, and $Z^{(k)}$.
     6) Calculate the reconstructed image $Y^{(k+1)}$ using (32) for RVIN and (33) for SPN.
     7) $\tau_1^{(k+1)}$=max$\left(p\cdot\tau_1^{(k)}, 0.45\right)$, $\tau_2^{(k+1)}$=max$\left(p\cdot\tau_2^{(k)}, 0.65\right)$.
Output: The final restored image $\hat{Y} = Y^{(k+1)}$
```

The proposed model in (11) will become much simpler when deal with SPN. For SPN corrupted images, there is no entry between 0 and 1 in the weight matrix W because each entry $w_{i,j} = 0$ or 1, then the proposed model in (11) is simplified into,

$$\begin{aligned}\hat{Y} = \arg\min_{Y,\alpha} \Bigg\{ &\sum_{i,j} \frac{1}{2} \| R_{i,j}(I - W) \otimes \left(R_{i,j}Y - D\alpha_{i,j}\right) \|_2^2 \\ &+ \sum_{i,j} \frac{1}{2} \| R_{i,j}W \otimes \left(R_{i,j}X - D\alpha_{i,j}\right) \|_2^2 \\ &+ \frac{\lambda_1}{2} \| W \otimes (Y - X) \|_2^2 \Bigg\} \quad s.t.\ \|\alpha_{i,j}\|_0 \le L\end{aligned}$$

After some similar manipulations, the reconstructed image can also be calculated pixel by pixel,

$$y_{i,j} = \begin{cases} x_{i,j}; & \text{if } w_{i,j} = 1 \\ \frac{z_{i,j}}{m_{i,j}}; & \text{if } w_{i,j} = 0 \end{cases} \tag{33}$$

From (32) and (33), one can see that in the (i, j)-th position without IN ($w_{i,j} = 1$), the pixel value is reasonably unchanged and this is crucial in preserving image details. On the other hand, if the (i, j)-th pixel is heavily corrupted ($w_{i,j} = 0$), the estimated value is calculated from the information around it. Finally, if the (i, j)-th position is just a slightly corrupted pixel ($0 < w_{i,j} < 1$), the estimated value is calculated from both the neighborhood suggested value and the noisy pixel itself, shrinking the neighborhood suggested value toward $x_{i,j}$.

E. Iterative Denoising Algorithm

For IN removal, we iteratively apply the WCSR-$l_2 l_1$ model to improve the denoising performance. That is, the output of current iteration is used as the input of next iteration. In each iteration, the weight matrix is updated via (2) for SPN and (3) for RVIN based on the current input image.

In our iterative denoising method, the initial image for Y in (12) is reconstructed from (4) and (5), and the initial dictionary is chosen as discrete cosine transform (DCT). The details of the proposed iterative IN removal algorithm is summarized in algorithm 1.

In algorithm 1, instead of updating the dictionary D each time, we choose to update it every q (e.g., $q = 2$) iterations to save the computation time. In step 7, the two thresholds

TABLE I

COMPARISON OF RESTORATION RESULTS IN PSNR AND SSIM FOR IMAGES CORRUPTED BY SPN

Image	30%				50%				80%			
	UTMF [38]	IBDND [39]	WESNR [29]	WCSR-l_2l_1	UTMF [38]	IBDND [39]	WESNR [29]	WCSR-l_2l_1	UTMF [38]	IBDND [39]	WESNR [29]	WCSR-l_2l_1
	PSNR											
lena	39.09	40.41	37.71	**42.69**	33.14	36.45	36.10	**38.95**	25.78	30.22	29.51	**31.06**
pepper	35.68	36.38	33.46	**37.35**	31.45	33.35	32.66	**34.30**	28.18	28.64	28.61	**29.41**
F16	35.65	36.86	35.57	**38.82**	30.57	33.21	32.77	**35.08**	27.69	27.62	26.88	**28.29**
house	42.25	43.26	41.65	**45.21**	34.85	40.53	40.25	**41.84**	30.46	33.43	34.59	**35.05**
barbara	28.14	29.32	30.12	**36.93**	25.32	26.63	28.59	**33.06**	22.25	22.89	23.73	**26.18**
boat	32.93	33.65	31.81	**36.55**	29.17	30.46	30.26	**32.90**	24.87	25.77	25.59	**26.68**
bridge	29.32	29.85	27.08	**30.49**	25.90	26.69	25.92	**27.55**	22.11	22.81	22.76	**23.36**
pentagon	32.49	32.89	30.09	**34.55**	29.27	30.03	29.18	**31.20**	25.33	25.82	**26.38**	26.19
average	34.44	35.33	33.44	**37.82**	29.96	32.17	31.97	**34.36**	25.83	27.15	27.26	**28.28**
	SSIM											
lena	0.9815	0.9823	0.9485	**0.9828**	0.9379	0.9646	0.9384	**0.9680**	0.8156	0.8997	0.8836	**0.9038**
pepper	0.9374	0.9457	0.8412	**0.9491**	0.8689	0.9001	0.8327	**0.9061**	0.8027	0.7827	0.7886	**0.8091**
F16	0.9766	0.9781	0.9762	**0.9785**	0.9337	0.9535	0.9176	**0.9573**	0.7637	**0.8815**	0.8588	0.8788
house	0.9807	0.9645	0.9695	**0.9851**	0.9514	**0.9853**	0.9626	0.9739	0.8921	0.9044	0.9318	**0.9385**
barbara	0.9047	0.9219	0.8594	**0.9672**	0.8106	0.8515	0.8311	**0.9353**	0.6253	0.6727	0.6673	**0.7853**
boat	0.9400	0.9463	0.8548	**0.9609**	0.8719	0.8964	0.8293	**0.9241**	0.7128	0.7530	0.7249	**0.7801**
bridge	0.9202	0.9288	0.7901	**0.9364**	0.8308	0.8582	0.7428	**0.8708**	0.5967	**0.6610**	0.5853	0.6572
pentagon	0.9316	0.9372	0.8299	**0.9522**	0.8575	0.8767	0.7980	**0.8985**	0.6676	0.6958	0.6740	**0.7026**
average	0.9466	0.9511	0.8837	**0.9639**	0.8828	0.9108	0.8566	**0.9293**	0.7408	0.7863	0.7643	**0.8069**

τ_1 and τ_2 are decreasing along with the increase of iteration number to detect more potential outliers [12], and p is the noise density estimated as $p = \frac{1}{N}\sum_{i,j}\left(1 - w_{i,j}\right)$.

III. SIMULATION RESULTS

In this section, experiments are conducted and compared with several existing methods to assess the noise reduction capability of the proposed algorithm. Simulations are carried out on various standard test images corrupted by RVIN or SPN. Fig. 2 shows some examples of the original noise-free images.

A. Implementation and Parameter Setting

Our proposed method can be implemented according to Algorithm 1. There are some parameters should be predetermined: two control parameters λ_1, λ_2, thresholds τ_1, τ_2, and the maximum iteration number k_{max}.

By our extensive experiments, the restoration results are not so sensitive to the two control parameters. The parameter pair chosen from $\lambda_1 \in [40, 65]$ and $\lambda_2 \in [560, 780]$ can achieve satisfactory results. Here, we set $\lambda_1 = 60$ and $\lambda_2 = 780$ in all our experiments. Experimentally, the initial values of τ_1 and τ_2 are set as $\tau_1 = 3$ and $\tau_2 = 3.8$ for all the tested images except for *barbara* image, which is rich in textures and requires relative larger thresholds. Hence, we set $\tau_1 = 4$ and $\tau_2 = 5$ for *barbara* images with low noise densities (e.g., 40% and 50%), while $\tau_1 = 3.5$ and $\tau_2 = 4.5$ for the high noise case (e.g., 60%).

The stopping criterion is very important for an iterative algorithm because iterative filtering should be stopped before it begins to severely destroy image details. We choose to stop the iterative filtering when

$$\frac{\|Y_{cur} - Y_{pre}\|_2^2}{\|Y_{pre}\|_2^2} \le \eta, \tag{34}$$

where $\eta = 0.003$ for SPN and 0.01 for RVIN; Y_{cur} and Y_{pre} represent the outputs of current and previous iterations, respectively. For the maximum iteration number, we set $k_{max} = 6$.

For sparse coding, the 8×8 patches with an overlap of 6 pixels between adjacent patches are extracted from the noisy image. The dictionary size is chosen as $D \in R^{64\times256}$, which means there are totally 256 atoms in the dictionary. The sparse ratio is set as $L = 5$.

B. Comparison of Image Restoration

The PSNR (peak signal to noise ratio) and SSIM (structural similarity) [41] are used to quantitatively evaluate the qualities of restored results. Generally, the higher PSNR and SSIM values indicate better qualities of the restored images.

For SPN removal, we compare our method with some recently developed methods, namely, UTMF [38], IBDND [39], and WESNR [29]. For RVIN reduction, the proposed method is compared with ACWM [5], Luo-iterative [6], CEF [7], ASWM [8], ROR-NLM [9], l_1-l_0 method [27], ROLD-EPR [12], and WESNR [29]. The codes of ROR-NLM, ROLD-EPR, and WESNR are provided by the authors, while other algorithms are implemented with the optimal parameters and iterations suggested by the original papers.

Table I and II list the PSNR and SSIM values from the methods for all the tested images with different noise densities about the SPN and RVIN, respectively. In both tables, the best values are marked in bold for convenient comparisons. From Table I, it is clear to observe that the proposed method achieves competitive or higher PSNR scores in comparison with the three state-of-the-art SPN removal methods. Though the SSIM values of our method are slightly less than the best ones in three cases (*house* image with 50% SPN, *F16* and *bridge* images with 80% SPN), our method outperforms other methods in all the rest cases.

TABLE II

COMPARISON OF RESTORATION RESULTS IN PSNR AND SSIM FOR IMAGES CORRUPTED BY RVIN

Image		PSNR									SSIM								
		ACWM [5]	Luo's [6]	ASWM [8]	l_1-l_0 [27]	CEF [7]	ROR-NLM [9]	ROLD-EPR [12]	WESNR [29]	WCSR-$l_2 l_1$	ACWM [5]	Luo's [6]	ASWM [8]	l_1-l_0 [27]	CEF [7]	ROR-NLM [9]	ROLD-EPR [12]	WESNR [29]	WCSR-$l_2 l_1$
lena	40%	29.58	30.77	32.29	33.23	32.11	32.97	32.72	33.37	**34.26**	0.8771	0.8807	0.9319	0.9377	0.9162	0.9359	0.9292	0.9096	**0.9442**
	50%	24.63	27.15	29.23	30.60	29.76	30.02	31.12	31.35	**32.16**	0.7648	0.7957	0.8823	0.9017	0.8850	0.8993	0.9015	0.8809	**0.9162**
	60%	20.40	22.61	25.04	27.02	25.90	25.60	28.98	26.57	**29.38**	0.5835	0.6302	0.7913	0.8430	0.7634	0.8102	0.8529	0.8331	**0.8640**
pepper	40%	28.89	29.53	30.28	31.05	30.27	31.24	31.11	31.12	**32.00**	0.8126	0.8137	0.8134	0.8639	0.8497	0.8641	0.8671	0.8227	**0.8829**
	50%	25.40	26.71	28.39	29.41	28.39	28.94	29.90	30.17	**30.55**	0.7069	0.7227	0.7709	0.8068	0.8101	0.8202	0.8271	0.8111	**0.8483**
	60%	21.53	23.53	25.35	27.04	25.35	25.85	28.19	26.63	**28.76**	0.5484	0.6113	0.7068	0.7606	0.7304	0.7452	0.7764	0.7510	**0.7990**
F16	40%	28.08	28.89	29.52	29.29	28.94	**29.68**	28.65	29.42	29.51	0.8595	0.8833	0.9148	0.8392	0.8970	**0.9184**	0.8879	0.9054	0.9159
	50%	24.46	26.62	27.05	26.97	27.03	27.54	27.54	28.24	**28.30**	0.7187	0.8008	0.8593	0.8595	0.8583	0.8827	0.8625	0.8862	**0.8881**
	60%	20.51	22.77	24.59	24.61	24.84	25.17	26.27	25.12	**26.88**	0.5083	0.6282	0.7664	0.8044	0.7339	0.8025	0.8063	0.7954	**0.8351**
house	40%	31.96	33.43	36.08	38.10	33.94	36.50	36.57	36.91	**38.63**	0.9098	0.9095	0.9276	0.9483	0.9380	0.9555	0.9586	0.9576	**0.9595**
	50%	27.27	29.72	34.62	35.21	31.22	33.35	34.44	35.36	**36.38**	0.7975	0.8359	0.9411	0.9069	0.9209	0.9327	0.9358	0.9369	**0.9402**
	60%	22.42	24.27	28.06	30.41	29.06	28.73	32.02	28.96	**33.08**	0.5938	0.6637	0.8593	0.9297	0.7991	0.8725	0.8985	0.8814	**0.8994**
barbara	40%	23.37	23.38	23.44	23.67	23.25	23.87	23.87	24.27	**26.44**	0.7158	0.7516	0.6709	0.6888	0.7027	0.7413	0.7397	0.7250	**0.8102**
	50%	22.17	22.19	22.97	23.48	23.02	23.27	23.36	23.63	**24.37**	0.6263	0.6663	0.6189	0.6640	0.6615	0.6866	0.6780	0.7019	**0.7040**
	60%	20.21	20.28	22.12	22.61	22.14	22.28	22.89	22.19	**23.08**	0.4925	0.5406	0.5713	0.6051	0.5933	0.6175	0.6309	0.6505	**0.6583**
boat	40%	26.74	26.88	27.27	26.89	27.08	27.59	27.04	27.41	**27.75**	0.8089	0.8025	0.8056	0.8274	0.8223	0.8328	0.8297	0.8019	**0.8561**
	50%	24.50	24.82	25.56	25.54	25.65	25.87	25.96	26.19	**26.40**	0.7062	0.7189	0.7258	0.7462	0.7658	0.7806	0.7807	0.7724	**0.8054**
	60%	21.45	21.62	23.80	23.89	24.16	24.17	25.01	24.38	**25.04**	0.5691	0.5675	0.6515	0.6685	0.6849	0.7083	0.7213	0.7020	**0.7388**
bridge	40%	23.52	23.59	23.97	23.58	23.85	24.18	24.51	24.37	**24.55**	0.7270	0.7388	0.6953	0.6899	0.7162	0.7254	**0.7563**	0.6920	0.7544
	50%	21.41	21.62	22.58	22.68	22.79	22.84	23.51	23.44	**23.69**	0.6262	0.6388	0.5830	0.5897	0.6419	0.6459	0.6873	0.6570	**0.6886**
	60%	19.12	19.17	21.11	21.03	21.41	21.19	22.52	21.49	**22.67**	0.4824	0.5005	0.4936	0.4789	0.5616	0.5462	0.6083	0.5658	**0.6104**
pentagon	40%	27.09	27.00	27.29	27.20	27.16	27.68	27.59	27.64	**27.99**	0.7838	0.7829	0.7640	0.7670	0.7801	0.7962	0.7940	0.7849	**0.8034**
	50%	25.47	25.33	26.19	26.11	26.24	26.56	26.65	26.73	**27.00**	0.7057	0.7047	0.6852	0.6771	0.7202	0.7354	0.7435	0.7254	**0.7535**
	60%	23.41	22.78	24.98	24.95	25.12	25.36	25.61	25.56	**25.89**	0.5898	0.5812	0.6178	0.6053	0.6572	0.6725	0.6762	0.6698	**0.6839**
average	40%	27.40	27.93	28.77	29.13	28.33	29.21	29.01	29.31	**30.14**	0.8118	0.8204	0.8204	0.8203	0.8278	0.8462	0.8453	0.8249	**0.8658**
	50%	24.41	25.52	27.07	27.50	26.76	27.30	27.81	28.14	**28.61**	0.7065	0.7355	0.7583	0.7690	0.7830	0.7979	0.8021	0.7965	**0.8180**
	60%	21.13	22.13	24.38	25.20	24.75	24.79	26.44	25.11	**26.85**	0.5460	0.5904	0.6823	0.7119	0.6905	0.7219	0.7463	0.7311	**0.7597**

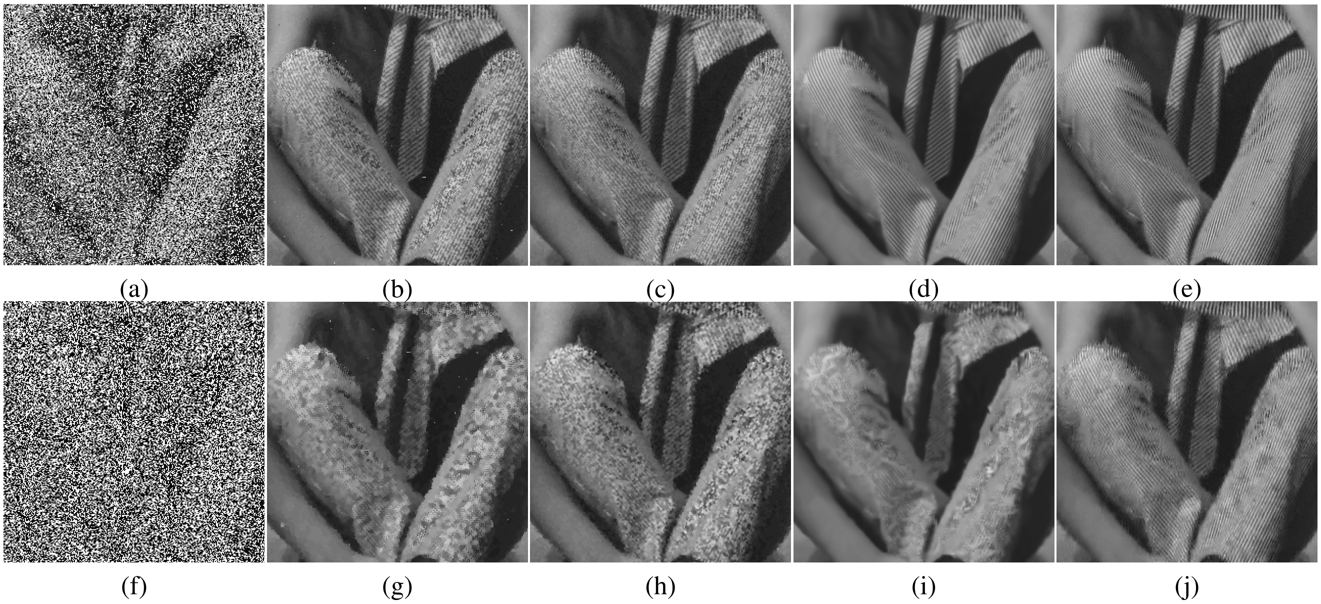


Fig. 3. Results of different algorithms in restoring the test *barbara* image corrupted by SPN with 50% and 80% noise densities respectively. Top row: 50% noise, bottom row: 80% noise. (a) Noisy image, (b) UTMF, (c) IBDND, (d) WESNR, (e) proposed; (f) Noisy image, (g) UTMF, (h) IBDND, (i) WESNR, (j) WCSR-$l_2 l_1$. Please zoom into pdf file for a detailed view.

From Table II, it is obvious to see that our proposed method generates the best results almost for all the tested images with different noise densities. Actually, in Table I and II, the proposed method obtains the highest averaged PSNR and SSIM values for all the tested images with different noise densities. This demonstrates that our method is more robust for images corrupted by IN with different noise levels.

The enlarged parts of recovered *barbara* images with 50% and 80% noise densities by all the tested methods are shown in Fig. 3 to give a visual impression.[1] From this figure, we can

[1] More restored results can be found in the website http://www.cis.umac.mo/cybernetics/WCSR-L2L1/DenoisedResults.htm. Matlab code available upon email request (yb27408@umac.mo or lichenghnu@gmail.com)

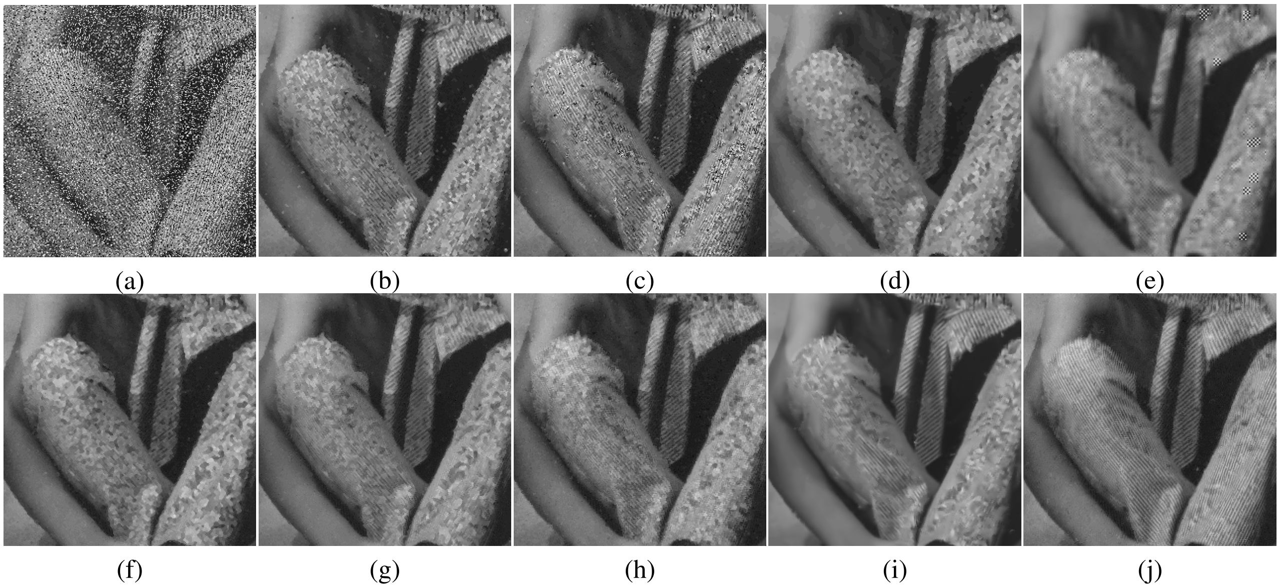


Fig. 4. Results of different algorithms in restoring *lena* image corrupted by 40% RVIN: (a) Noisy image, (b) ACWM, (c) Luo', (d) ASWM, (e) $l_1 - l_0$ (f) CEF; (g) ROR-NLM, (h) ROLD-EPR, (i) WESNR, (j) WCSR-l_2l_1. Please zoom into pdf file for a detailed view.

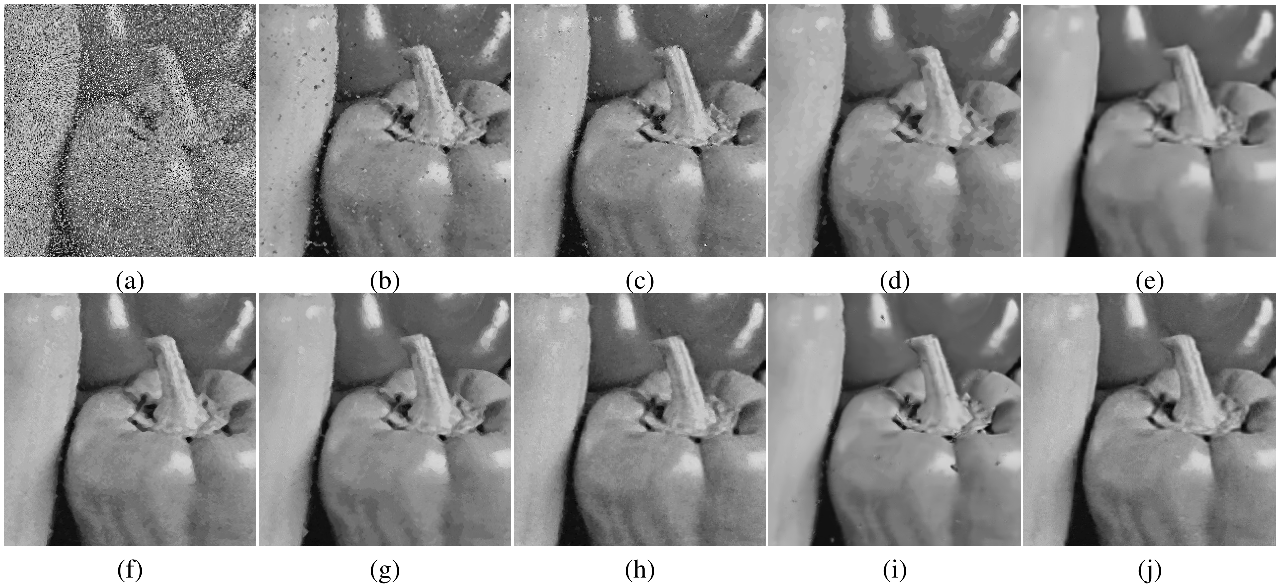
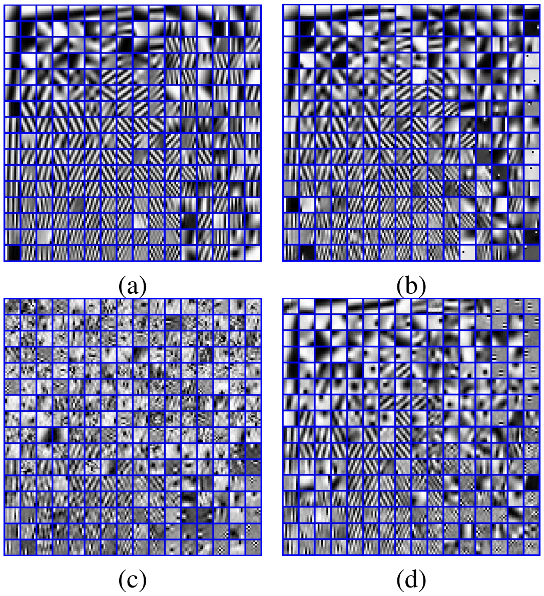


Fig. 5. Results of different algorithms in restoring *pepper* image corrupted by 50% RVIN: (a) Noisy image, (b) ACWM, (c) Luo', (d) ASWM, (e) $l_1 - l_0$ (f) CEF; (g) ROR-NLM, (h) ROLD-EPR, (i) WESNR, (j) WCSR-l_2l_1. Please zoom into pdf file for a detailed view.

see that the SR based methods outperform filter based methods in reduction of SPN. The UIMF and IBDND methods not only cannot completely remove the SPN, but also bring in many artifacts. On the contrary, the WESNR produces much better results. However, it still lost lots of image details and destroy the edges, especially when the noise density is high, where the number of clean pixels can be used for sparse coding is limited. The results generated by the proposed method have very good visual qualities. Even though the noise density is 80%, our method still can remove almost all the SPN and preserve most of the image details.

Fig. 4 and 5 list the enlarged parts of the restored *barbara* and *pepper* images corrupted by 40% and 50% RVIN, respectively. From these two figures, we can see that other methods, except for l_1-l_0, ROR-NLM, ROLD-EPR, and WESNR, show very bad performance in suppressing RVIN. By comparison, ROR-NLM generates relative better results, nevertheless, it blurs the outputs and brings in some artifacts. Though l_1-l_0 and WESNR can removal most of noise, the results are somehow over filtered. This is because none of them takes into account the IN characteristics in sparse coding or reconstruction. The results produced by ROLD-EPR method verify that this method indeed has a good capability of preserving image edges. However, there are still some noticeable noise unremoved due to the imperfect filtering technique it used. In contrast, the results generated by our method show very good visual qualities, which illustrates that the proposed method is very effective in reduction of IN. Both the noise


(a) (b)

(c) (d)

Fig. 6. The trained dictionaries by different methods: The first column: trained dictionaries for *barbara* image with 50% SPN (shown in Fig. 3(a)). The second column: trained dictionaries for *barbara* image with 40% RVIN (shown in Fig. 4(a)). (a) the proposed dictionary learning method, (b) the proposed dictionary learning method, (c) K-SVD, (d) K-SVD.

are removed thoroughly and the image details (e.g., edges and textures) are well preserved. Especially for the images rich in textures (e.g., *barbara* image in Fig. 4), the superiority our method is more obvious.

Fig. 6 shows the dictionaries trained by our proposed dictionary learning method and the K-SVD [34] algorithm for the *barbara* image corrupted by 50% SPN and 40% RVIN, respectively. From this figure, one can see that, compared with those trained from K-SVD, the two dictionaries learned by the proposed method are less noisy and contains more features of *barbara* image (e.g., the texture), hence they are more appropriate to reconstruct the original image.

C. Effectiveness of the L_2-Norm and L_1-Norm

In the proposed model, the l_2-norm is used to make the reconstructed pixels approach to the corresponding clean ones to preserve more image details. The l_1-norm is designed to regularize the IN corrupted pixels, which improves the robustness of the final model. Both of them are very important in the proposed WCSR-l_2l_1 model.

To further verify the effectiveness of the l_1-norm and l_2-norm in (11). We implement two variants of the final denoising model. Let the "WCSR-l_2" denote the IN removal method using SR and l_2-norm, which solves the following minimization problem,

$$(\hat{\alpha},\hat{D},\hat{Y}) = \arg\min_{\alpha,D,Y}\left\{\sum_{i,j}\frac{1}{2}\|R_{i,j}W\otimes\left(R_{i,j}X-D\alpha_{i,j}\right)\|_2^2 + \sum_{i,j}\frac{1}{2}\|R_{i,j}(I-W)\otimes\left(R_{i,j}Y-D\alpha_{i,j}\right)\|_2^2 + \frac{\lambda_1}{2}\|W\otimes(Y-X)\|_2^2\right\}, \quad s.t.\ \|\alpha_{i,j}\|_0\le L.$$

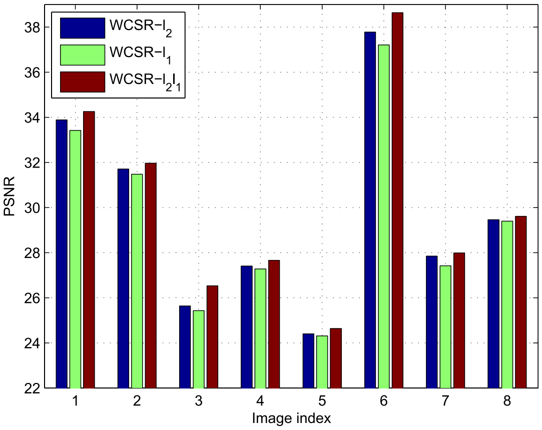


Fig. 7. PSNR values of recovered images by using WCSR-l_2, WCSR-l_1, and WCSR-l_2l_1. The x-coordinate denotes the image index shown in Fig. 2: 1: *lena*, 2: *pepper*, 3: *barbara*, 4: *boat*, 5: *bridge*, 6: *house*, 7: *pentagon*, 8: *F16*.

The other variant denoted by "WCSR-l_1" intends to remove IN by minimizing

$$(\hat{\alpha},\hat{D},\hat{Y}) = \arg\min_{\alpha,D,Y}\left\{\sum_{i,j}\frac{1}{2}\|R_{i,j}W\otimes\left(R_{i,j}X-D\alpha_{i,j}\right)\|_2^2 + \sum_{i,j}\frac{1}{2}\|R_{i,j}(I-W)\otimes\left(R_{i,j}Y-D\alpha_{i,j}\right)\|_2^2 + \lambda_2\|\bar{W}\otimes(Y-X)\|_1\right\}, \quad s.t.\ \|\alpha_{i,j}\|_0\le L.$$

The two variants can be implemented by modifying Algorithm 1 slightly. They were then used to recover the images shown in Fig. 2 corrupted by 40% RVIN. The restored results were compared to those of WCSR-l_2l_1 and the PSNR values were plotted in Fig. 7.

From Fig. 7, one can see that WCSR-l_2 gains higher PSNR scores than WCSR-l_1, which indicates l_2-norm plays a more important role compared to l_1-norm in the final model. This is because that l_2-norm forces the reconstructed pixels approach to the clean ones, which preserves more details of the original image, while l_1-norm just makes the model robust to outliers. In contrast, the WCSR-l_2l_1 obtains the best performance than the WCSR-l_2 and WCSR-l_1. This means that both the l_2-norm and l_1-norm in the final model are necessary and meaningful. Without any of them, the denoising performance of the final model will be weakened.

D. Running Time

The main computation cost of our proposed algorithm comes from the sparse coding and dictionary learning subproblems. The reason is that both these two subproblems adopt the OMP algorithm to calculate the coefficients and the OMP is time-consuming. As a result, our method is somehow slower than some other two-phase methods [9], [12]. More specifically, for the 256 × 256 *house* image, it costs about 48 seconds CPU time for each outer iteration of our proposed method by using our unoptimized Matlab codes on a computer equipped with 3.40 GHz CPU.

It is worth to note that there are several ways to speed up the proposed algorithm. On the one hand, the fast split or greedy algorithms can be used instead of OMP to solve the l_0 minimization problem more effectively. On the other hand, the proposed method can be parallelized and run on a more powerful GPU machine to further reduce the running time.

IV. Conclusion

In this paper, we propose a weighted couple sparse representation based algorithm to remove IN in images. In summary, a weight matrix is introduced to describe the cleanliness of a pixel and to determine the contribution of each pixel offered in the sparse coding stage. The noisy and reconstructed images are coupled in the SR model where the reconstructed image is used to compensate the lost information in the noisy image and to make the coefficients more appropriate to reconstruct the noise-free image. To further improve the denoising performance, the pixels are classified into three categories, and different types of pixels are assigned with different regularizations. Besides, the dictionary is trained directly on the raw data to capture more features of the original image. Extensive experiments demonstrate that the proposed method achieves better performance in reduction of IN compared with several state-of-the-art denoising algorithms with respect to both the quantitative measurements and the visual effects.

It is worth to extend the proposed method for color image restoration, considering the correlations among R (red), G (green), and B (blue) channels. Besides, integrating the non-local self similarity priors [42], [43] into the weighted SR model may preserve more image details and texture information. This can be a good direction of future work to produce much higher quality outputs.

Appendix
Proof of Property 1

Proof: The dictionary atom updating is reformulated as

$$\hat{d} = \arg\min_{d} F(d, \alpha) \tag{35}$$

This can be achieved by calculating the derivative of $F(d)$ with respect to d, and set it to 0.

$$\frac{\partial F(d, \alpha)}{\partial d} = (W \otimes W) \otimes (E - d\alpha)\alpha^T = 0 \tag{36}$$

(36) leads to

$$\tilde{W} \otimes (d\alpha)\alpha^T = \tilde{W} \otimes E\alpha^T \tag{37}$$

where $\tilde{W} = W \otimes W$, and the superscript T denotes the transposition. The left side of (37) can be rewritten as

$$\begin{aligned} \tilde{W} \otimes (d\alpha)\alpha^T &= \begin{pmatrix} \sum_{j=1}^{N} \alpha_j^2 \tilde{\omega}_{1,j} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sum_{j=1}^{N} \alpha_j^2 \tilde{\omega}_{n,j} \end{pmatrix} \cdot \begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix} \\ &= \Phi \cdot d \end{aligned} \tag{38}$$

where $\Phi = \text{diag}(W(\alpha \otimes \alpha)^T)$ is a diagonal matrix.

Substituting (38) into (37), one can obtain

$$\hat{d} = \Phi^{-1} \cdot (\tilde{W} \otimes E\alpha^T) \tag{39}$$

After the atom d has been updated, the coefficient can then be updated by the similar way, calculating the derivative of $F(d, \alpha)$ with respect to α and setting it to 0, as follows,

$$\frac{\partial F(d, \alpha)}{\partial \alpha} = d^T \cdot \left(\tilde{W} \otimes (E - d\alpha)\right) = 0 \tag{40}$$

in which $\tilde{W}$ is defined the same in (37). (40) leads to

$$d^T(\tilde{W} \otimes (d\alpha)) = d^T(\tilde{W} \otimes E) \tag{41}$$

After some manipulations, the left side in (41) can be rewritten as,

$$\begin{aligned} d^T(\tilde{W} \otimes (d\alpha)) &= \left(\sum_{i=1}^{n} \tilde{\omega}_{i1} d_i^2 \alpha_1, \cdots, \sum_{i=1}^{n} \tilde{\omega}_{iN} d_i^2 \alpha_N, \right) \\ &= \alpha \cdot \Psi \end{aligned} \tag{42}$$

where $\Psi = \text{diag}\left(\sum_{i=1}^{n} \tilde{\omega}_{i1} d_i^2, \cdots, \sum_{i=1}^{n} \tilde{\omega}_{iN} d_i^2\right) = \text{diag}\left(\tilde{W}^T \cdot (d \otimes d)\right)$ is a diagonal matrix. Substituting (42) into (41), then α is updated as,

$$\hat{\alpha} = d^T(\tilde{W} \otimes E) \cdot \Psi^{-1} \tag{43}$$

■

Acknowledgment

The authors would like to thank the associate editor and anonymous reviewers for their valued comments and constructive criticisms which are very helpful to improve the quality of this paper. They thank Prof. Dong and Prof. Chan for supplying the code of their algorithm in [12], Dr. Xiong for providing the MATLAB implementation of the method in [9], and Dr. Jiang and Prof. Zhang for sharing their code in [29]. They also would like to thank Prof. Pun Chi Man for his help of editing this paper.

References

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2008.

[2] T. Nodes and N. C. Gallagher, Jr., "Median filters: Some modifications and their properties," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 30, no. 5, pp. 739–746, Oct. 1982.

[3] D. R. K. Brownrigg, "The weighted median filter," *ACM Commun.*, vol. 27, no. 8, pp. 807–818, 1984.

[4] 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.

[5] 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.

[6] W. Luo, "A new efficient impulse detection algorithm for the removal of impulse noise," *IEICE Trans. Fund. Electron., Commun. Comput. Sci.*, vol. 88, no. 10, pp. 2579–2586, 2005.

[7] U. Ghanekar, A. K. Singh, and R. Pandey, "A contrast enhancement-based filter for removal of random valued impulse noise," *IEEE Signal Process. Lett.*, vol. 17, no. 1, pp. 47–50, Jan. 2010.

[8] S. Akkoul, R. Ledee, R. Leconge, and R. Harba, "A new adaptive switching median filter," *IEEE Signal Process. Lett.*, vol. 17, no. 6, pp. 587–590, Jun. 2010.

[9] B. Xiong and Z. 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.

[10] L. Liu, C. L. P. Chen, Y. Zhou, and X. You, "A new weighted mean filter with a two-phase detector for removing impulse noise," *Inf. Sci.*, vol. 315, pp. 1–16, Sep. 2015.

[11] 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.

[12] Y. Dong, R. H. Chan, and S. Xu, "A detection statistic for random-valued impulse noise," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 1112–1120, Apr. 2007.

[13] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, 2009.

[14] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[15] W. Dong, X. Li, D. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 457–464.

[16] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[17] J. Jiang, R. Hu, Z. Wang, and Z. Han, "Face super-resolution via multilayer locality-constrained iterative neighbor embedding and intermediate dictionary learning," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4220–4231, Oct. 2014.

[18] J. Jiang, R. Hu, Z. Wang, and Z. Han, "Noise robust face hallucination via locality-constrained representation," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1268–1281, Aug. 2014.

[19] L. Ma, J. Yu, and T. Zeng, "Sparse representation prior and total variation–based image deblurring under impulse noise," *SIAM J. Imag. Sci.*, vol. 6, no. 4, pp. 2258–2284, 2013.

[20] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3336–3351, 2014.

[21] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and M. Yi, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[22] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 625–632.

[23] J. Wang, C. Lu, M. Wang, P. Li, S. Yan, and X. Hu, "Robust face recognition via adaptive sparse representation," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2368–2378, 2014.

[24] S. Li, L. Fang, and H. Yin, "An efficient dictionary learning algorithm and its application to 3D medical image denoising," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 2, pp. 417–427, Feb. 2012.

[25] S. Yang, W. Min, L. Zhao, and Z. Wang, "Image noise reduction via geometric multiscale ridgelet support vector transform and dictionary learning," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4161–4169, Nov. 2013.

[26] R. Yan, L. Shao, and Y. Liu, "Nonlocal hierarchical dictionary learning using wavelets for image denoising," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4689–4698, Dec. 2013.

[27] Y. Xiao, T. 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, 2011.

[28] J. Liu, X.-C. Tai, H. Huang, and Z. Huan, "A weighted dictionary learning model for denoising images corrupted by mixed noise," *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 1108–1120, Mar. 2013.

[29] J. Jiang, L. Zhang, and J. Yang, "Mixed noise removal by weighted encoding with sparse nonlocal regularization," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2651–2662, Jun. 2014.

[30] Z. Zhou, "Cognition and removal of impulse noise with uncertainty," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3157–3167, Jul. 2012.

[31] H. Hwang and R. Haddad, "Adaptive median filters: New algorithms and results," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 499–502, Apr. 1995.

[32] L. Liu, C. L. P. Chen, Y. Zhou, and Y. Y. Tang, "Impulse noise removal using sparse representation with fuzzy weights," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2014, pp. 4052–4057.

[33] M. Nikolova, "A variational approach to remove outliers and impulse noise," *J. Math. Imag. Vis.*, vol. 20, nos. 1–2, pp. 99–120, 2004.

[34] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[35] N. Srebro and T. Jaakkola, "Weighted low-rank approximations," in *Proc. ICML*, vol. 3. 2003, pp. 720–727.

[36] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[37] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," *Neural Parallel Sci. Comput.*, vol. 11, no. 4, pp. 351–368, 2003.

[38] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, and C. H. PremChand, "Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter," *IEEE Signal Process. Lett.*, vol. 18, no. 5, pp. 287–290, May 2011.

[39] I. F. Jafar, R. A. AlNa'mneh, and K. A. Darabkh, "Efficient improvements on the BDND filtering algorithm for the removal of high-density impulse noise," *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 1223–1232, Mar. 2013.

[40] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[41] Z. Wang, A. C. Bovik, H. R. Sheikh, 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.

[42] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Jun. 2005, pp. 60–65.

[43] J. Zhang, D. Zhao, R. Xiong, S. Ma, and W. Gao, "Image restoration using joint statistical modeling in a space-transform domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 915–928, Jun. 2014.

Chun Lung Philip Chen (S'88–M'88–SM'94–F'07) received the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, in 1985, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1988.

He worked at U.S. for 23 years as a Tenured Professor, as a Department Head, and the Associate Dean in two different universities. He is currently the Dean of the Faculty of Science and Technology, University of Macau, Macau, China, and a Chair Professor of the Department of Computer and Information Science.

Dr. Chen's research areas are systems, cybernetics, and computational intelligence. He is a fellow of the American Association for the Advancement of Science. After being the IEEE SMC Society President (2012-2013), he is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS (2014-present) and an Associate Editor of several IEEE TRANSACTIONS. He is also the Chair of TC 9.1 Economic and Business Systems of IFAC. In addition, he is a Program Evaluator of the Accreditation Board of Engineering and Technology Education in Computer Engineering, Electrical Engineering, and Software Engineering Programs.

Licheng Liu received the B.S. degree in information and computational science from the China University of Geosciences, Wuhan, China, in 2010, and the M.S. degree in applied mathematics from Hunan University, Changsha, China, in 2012. He is currently pursuing the Ph.D. degree with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China.

His research interests include image processing, sparse representation, pattern recognition, and machine learning.

Long Chen (M'11) received the B.S. degree in information sciences from Peking University, Beijing, China, in 2000, the M.S.E. degree from the Institute of Automation, Chinese Academy of Sciences, in 2003, the M.S. degree in computer engineering from the University of Alberta, Canada, in 2005, and the Ph.D. degree in electrical engineering from The University of Texas at San Antonio, USA, in 2010. From 2010 to 2011, he was a Post-Doctoral Fellow with The University of Texas at San Antonio. He is currently an Assistant Professor with the Department of Computer and Information Science, University of Macau, China.

His current research interests include computational intelligence, Bayesian methods, and other machine learning techniques and their applications. He has been working in publication matters for many IEEE conferences, and was the Publication Co-Chair of the IEEE International Conference on Systems, Man and Cybernetics in 2009, 2012, and 2014.

Yuan Yan Tang (F'04) is currently a Chair Professor with the Faculty of Science and Technology, University of Macau, and a Professor/Adjunct Professor/Honorary Professor at several institutes, including Chongqing University, China, Concordia University, Canada, and Hong Kong Baptist University, Hong Kong. He has authored over 400 academic papers, and has authored or co-authored over 25 monographs/books/bookchapters. His current interests include wavelets, pattern recognition, and image processing. He is the Founder and the Editor-in-Chief of International Journal on Wavelets, Multiresolution, and Information Processing, and an Associate Editor of several international journals. He is the Founder and Chair of Pattern Recognition Committee in the IEEE SMC. He has served as the General Chair, Program Chair, and Committee Member for many international conferences. He is the Founder and General Chair of the series International Conferences on Wavelets Analysis and Pattern Recognition. He is the Founder and the Chair of the Macau Branch of International Associate of Pattern Recognition. He is the fellow of the International Association for Pattern Recognition.

Yicong Zhou (M'07–SM'14) received the B.S. degree from Hunan University, Changsha, China, in 1992, and the M.S. and Ph.D. degrees from Tufts University, MA, USA, in 2008 and 2010, respectively, all in electrical engineering.

He is currently an Assistant Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests focus on multimedia security, image/signal processing, pattern recognition, and medical imaging. He is a member of the International Society for Photo-Optical Instrumentations Engineers and Association for Computing Machinery. He won the third price of Macau Natural Science Award in 2014.