Restoration of images corrupted by impulse and mixed Gaussian-impulse noise by iterative mixed soft-hard thresholding

Marko Filipović
Rudjer Bošković Institute
Bijenička 54, 10000 Zagreb, Croatia
filipov@irb.hr

A. Jukić
Department of Medical Physics and Acoustics
University of Oldenburg
26111 Oldenburg, Germany
ante.jukic@uni-oldenburg.de

June 17, 2014

Abstract

We address the problem of restoration of images which have been affected by impulse or a combination of impulse and Gaussian noise. We propose a patch-based approach that exploits approximate sparse representation of image patches in learned dictionary. For every patch, sparse representation in learned dictionary is enforced by ℓ_1-norm penalty, and sparsity of the residual is enforced by ℓ_0-quasi-norm penalty. The obtained non-convex problem is solved iteratively by a combination of soft and hard thresholding, and a proof of convergence to a local minimum is given. Experimental evaluation suggests that the proposed approach can produce state-of-the-art results for some types of images, especially in terms of the structural similarity (SSIM) measure.

1 Introduction

Image denoising is a fundamental task in image processing. In this paper, we are interested in a special kind of noise, which is a mixture of impulse and Gaussian noise. Generally, several types of noise affect images: Poissonian, Gaussian and impulse. Poisson-Gaussian noise mixture can generally (when no data is lost, i.e. an image is not affected by impulse noise) be transformed into pure Gaussian noise [10], or can be treated by specialized methods [13]. However, the mixture of Gaussian and impulse noise is quite challenging for denoising. Two models of impulse noise that are used in the literature are salt-and-pepper noise and random-valued impulse noise. Let us suppose that the dynamic range of an image is $[d_{\min}, d_{\max}]$. Then, in the salt-and-pepper noise model, every image pixel is replaced, with a given probability, with a value $d_{\min}$ or $d_{\max}$. The second model of impulse noise, which is much more difficult to handle, assumes that a pixel value is replaced, again with a given probability, with a random value in the range $[d_{\min}, d_{\max}]$. Since detecting salt-and-pepper noise is much easier than random-valued impulse noise, we concentrate on the more general scenario of random-valued impulse noise in this paper.

Before introducing and describing our approach, we review previous work.

Most approaches for removing a mixture of Gaussian and impulse noise generally start by detecting the pixels corrupted by impulse noise. Then, after the influence of these noisy pixels is reduced, some robust noise removal method is used. In [11], a Rank Order Absolute Differences (ROAD) detector was used for detection of noisy pixels, while a trilateral filter was used for image restoration. Similar approach was presented in [8], where an improvement of the ROAD detector was proposed, and a variational method was used for restoration of corrupted pixels.

In [22], a three-phase denoising approach was proposed: first, the outlier candidates pixels are detected using a median-type filter; then, the initial approximation of a clean image is computed by a variant of the KSVD denoising algorithm [9, 16] using only pixels that are declared as clean; then, a model enforcing small ℓ_1-norm of the error and sparse representation (measured by ℓ_0) of image patches in learned dictionary is approximately solved. We describe the model used in that paper in more details in Section 2.3 since it is similar to the one used in this paper.

Similar approaches were presented in [5, 4, 14], where a median-type filter is used for detection of noisy pixels, and then a variational approach (based on total variation or ℓ_1 norm of framelet coefficients) is used for image restoration. These papers simultaneously tackle the problem of blur in images (as noted in [7], this makes the problem of detection of noisy pixels even easier, since the image is smoother). Therefore, the framework considered in these papers is somewhat different from the one discussed here.

In [7], a patch-based approach for removing impulse noise, based on robust statistics, was proposed. This approach has three main steps: first, the ratio of (impulse) noise-corrupted pixels is estimated using some impulse noise detector; then, for every patch a set of similar patches is found using some robust measure of similarity between patches; then, an estimate of clean patch is found using robust statistical estimation on the set of similar patches found in the first step. This approach provided state-of-the-art results, as demonstrated in [7]. We perform extensive experimental comparison both with the approach in [7] and the one in [22] in Section 3.

In [23], a model similar to the one proposed here was used. Namely, ℓ_0 penalty on the noise was used, while the total variation term was used as a regularization (therefore, a global regularization was used, unlike patch-based, as used in this paper). Alternating minimization with respect to the image approximation and the set of noise-free pixels was used for solving the resulting model, and a convergence of the algorithm to a coordinatewise minimum, i.e. a local minimum (with some minor modification of the algorithm) was proven. In [23], the term "blind inpainting" was used, although it generally refers to problems with structured patterns of corrupted pixels, which is not the case considered in that paper. Recovering large structured patterns of corrupted pixels is more difficult problem setting than the one considered here and in all mentioned papers.

Another recent method for both removal of impulsive noise and blind inpainting (although only with "easy" pattern of missing pixels: lines) was described in [19]. They used structured sparsity as a regularization by adapting the approach from [24] to impulsive noise removal and blind inpainting. Their algorithm first initializes a list of subspaces, and the indices of estimated corrupted pixels. The first step finds the nearest subspace to each image patch, ignoring the entries suspected to be corrupted. The second step computes a new basis for each cluster by the variant of robust principal component analysis (RPCA) algorithm. The basis for each subspace is interpreted as a local dictionary for similar patches, but since it is constrained to be orthogonal, patch denoising is easy given the basis for its corresponding subspace. Therefore, their approach is a variant of the K-subspaces method. We refer the reader to the paper [24] for further details.

In [15] a method for image deblurring under impulse noise was presented. The authors proposed a patch-based model with three terms: the sparse representation prior, the total variation regularization, and the data-fidelity term. They also used a two-phase approach for image restoration: identification of the possible impulse noise positions, followed by patch-based image recovery based on noise position information. Alternating minimization was used to solve the model. Good performance of the proposed method was demonstrated. We note that a conceptually similar approach was proposed in [18], however it is computationally demanding.

Yet another approach for general inverse problems in imaging was proposed in recent paper [12]. It uses a combined image regularization model based on total generalized variation and shearlets. Their approach, as demonstrated in the paper, recovers both edges and fine details much better than the regularization models based on total variation and wavelets. The proposed model is solved by variable splitting and the alternating direction method of multipliers. Also, the convergence of the algorithm is presented. Computational complexity of the algorithm wasn't discussed in much detail, similarly to all the approaches discussed in this section.

In this paper, an approach for removing impulse or a combination of impulse and Gaussian noise is proposed. For each patch, an optimization problem is formulated, where a sparse representation in a learned dictionary is enforced by ℓ_1-norm penalty, and a sparse residual is enforced by ℓ_0-quasi-norm penalty. The obtained non-convex problem is solved in an iterative procedure by a combination of soft and hard thresholding, with established convergence to a local minimum. In the experimental section a detailed comparison against two state-of-the-art methods is performed. Obained results show that the proposed method produces good results for some types of images, especially in terms of the structural similarity (SSIM) measure.

1.1 Notation and organization of the paper

Scalars are denoted by lowercase, vectors by bold lowercase, and matrices by bold uppercase letters. Operators and sets are denoted by uppercase or calligraphic letters. Transpose of matrix $\mathbf{D}$ is denoted by $\mathbf{D}^T$. The estimation of $\mathbf{x}$ at iteration k is denoted by $\mathbf{x}^{(k)}$. Componentwise multiplication (for vectors and matrices) is denoted by $\otimes$.

In Section 2 we describe our approach. Extensive numerical experiments and comparison with state-of-the-art methods are presented in Section 3. Conclusions are given in Section 4.

2 Main

The overall process of denoising consists of several steps: (offline) dictionary learning, impulse detection and alternating minimization algorithm; these are described in the following subsections.

2.1 Dictionary learning

In [22], dictionary for sparse representation was learned on damaged image itself, using a variant of the K-SVD algorithm. However, in this paper we simply learn the dictionary offline, on a training set of images of natural scenes. The dictionary is then fixed during the image restoration process. The dictionary is learned using the independent component analysis (ICA) on the set of patches extracted from the images in the training set. We demonstrate that this approach gives good results. Similar results could be obtained using some other dictionary learning algorithm (for example, K-SVD, or ℓ_1-based dictionary learning from [17]), but the dictionary learned using ICA consistently gave better results in terms of the SSIM (Structural SIMilarity) measure (described later), as demonstrated in the Experiments section.

2.2 Impulse detection

We use the well known ROAD detector [11], as also used in [7]. It can be described as follows. For every image pixel, all absolute differences between that pixel and pixels in a surrounding patch are computed. These differences are then sorted, and the value of ROAD statistic at that pixel is obtained by computing the sum of 4 smallest differences. When ROAD is above some threshold, the pixel is considered as noisy (because ROAD measures how close is the value of image pixel to the values in a patch around it). Parameters used in experiments are described in Section 3.

2.3 Image restoration algorithm

We propose the following formulation of the problem. For every image patch, the following problem is solved:

$$\min_{\mathbf{x},\,\mathbf{f}} \frac{1}{2} \left\|\mathbf{\Omega} \otimes (\mathbf{u} - \mathbf{D}\mathbf{x} - \mathbf{f})\right\|_2^2 + \lambda_1 \left\|\mathbf{x}\right\|_1 \quad \text{subject to} \quad \left\|\mathbf{\Omega} \otimes \mathbf{f}\right\|_0 \leq \lambda_2. \tag{1}$$

Here, $\mathbf{u}$ denotes a patch of the noisy image (therefore, generally it is affected by both Gaussian and impulse noise), $\mathbf{\Omega}$ is the matrix indicating the positions of clean pixels in current patch (as indicated by the ROAD detector), and λ_1, λ_2 are parameters. λ_1 is a parameter that depends on the level of Gaussian (bounded) noise, while λ_2 controls the level of impulsive (sparse) noise. Unconstrained formulation of the problem is also possible, wherein the regularization term $\lambda_2 \left\|f\right\|_0$ is added to the objective function in (1) (in that case, λ_2 has different interpretation). However, it seems more natural to consider the formulation (1) because here λ_2 has simple interpretation: it bounds the number of pixels affected by impulsive noise. It should be noted that another approach would be to use a *global* regularization on f (since the patches are generally overlapping). However, the local approach used here is simpler and gives good results. Among many possible formulations, it is also possible to use ℓ_1 norm as a measure of sparsity of noise; however, in our simulations better results were obtained with (1).

Similar model was used in the paper [22]

$$\min_{\mathbf{x},\,\mathbf{D},\,\boldsymbol{\alpha}_{ij}} \lambda \left\|\mathbf{\Omega} \otimes (\mathbf{x} - \mathbf{u})\right\|_2^2 + \beta \left\|(\mathbf{1}_\mathbf{u} - \mathbf{\Omega}) \otimes (\mathbf{x} - \mathbf{u})\right\|_1 + \\ + \sum_{(i,\,j)\in\mathcal{P}} \left\|\mathbf{D}\boldsymbol{\alpha}_{ij} - \mathcal{R}_{ij}\mathbf{x}\right\|_2^2 + \sum_{(i,\,j)\in\mathcal{P}} \mu_{ij} \left\|\boldsymbol{\alpha}_{ij}\right\|_0. \tag{2}$$

Here, $\mathbf{1}_{\mathbf{u}}$ is the matrix of ones of the same size as $\mathbf{u}$, $\mathcal{R}_{ij}$ is the operator that extracts image patch at position (i, j), and λ, β and μ_{ij} are parameters. In other words, model in (2) uses ℓ_1 norm for the residual $(u - \tilde{u})$ and ℓ_0 "norm" for the coefficients of image patches in the dictionary D. The first term in (2) allows some Gaussian (bounded) noise too, and the parameter λ should be inversely proportional to the level of Gaussian noise. The formulation (2) combines local and global regularization of the denoising problem, by combining sparse representation of image patches and ℓ_1-norm of the image residual.

Formulation (1) can be solved by a combination of soft and hard thresholding. Namely, let us suppose that the norm of matrix $\mathbf{D}$ is less than 1 (since we are working on a patch level, $\mathbf{D}$ is a small matrix and it can be normalized if necessary, so that this assumption is realistic). Then, the objective function in (1) can be majorized for every $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{f} \in \mathbb{R}^m$ as follows [2]:

$$\frac{1}{2}\left\|\mathbf{u}-\mathbf{D}\mathbf{x}-\mathbf{f}\right\|_2^2 \quad \leq \quad \frac{1}{2}\left(\left\|\mathbf{u}-\mathbf{D}\mathbf{x}-\mathbf{f}\right\|_2^2 + 2\left\|\mathbf{x}-\mathbf{v}\right\|_2^2 + 2\left\|\mathbf{f}-\mathbf{w}\right\|_2^2 - \left\|\mathbf{D}\mathbf{x}+\mathbf{f}-\mathbf{D}\mathbf{v}-\mathbf{w}\right\|_2^2\right). \quad (3)$$

$\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{w} \in \mathbb{R}^m$ are arbitrary here (in the algorithm, we set $\mathbf{v} = \mathbf{x}^{(k)}$, $\mathbf{w} = \mathbf{f}^{(k)}$). Also, we have neglected componentwise multiplication with $\mathbf{\Omega}$ in (1) for simplicity. The same majorization is valid on the set indicated by $\mathbf{\Omega}$ too. After substitution of the above majorization into (1) and straightforward calculation, we arrive at the following problem:

$$\begin{aligned}\min_{\mathbf{x},\,\mathbf{f}} \frac{1}{2}\left\|\mathbf{x}-\left(\mathbf{v}+\frac{1}{2}\mathbf{D}^T\left(\mathbf{\Omega}\otimes(\mathbf{u}-\mathbf{D}\mathbf{v}-\mathbf{w})\right)\right)\right\|_2^2 + \lambda_1\left\|\mathbf{x}\right\|_1 + &\\ + \frac{1}{2}\left\|\mathbf{f}-\left(\mathbf{w}+\frac{1}{2}\mathbf{\Omega}\otimes(\mathbf{u}-\mathbf{D}\mathbf{v}-\mathbf{w})\right)\right\|_2^2 \quad \text{subject to} \quad \left\|\mathbf{\Omega}\otimes\mathbf{f}\right\|_0 \leq \lambda_2. & \quad (4)\end{aligned}$$

The components of $\mathbf{x}$ and $\mathbf{f}$ in (4) are decoupled, and minimization can be performed separately for $\mathbf{x}$ and $\mathbf{f}$. Also, each minimization sub-problem is easy and has closed-form solution. Namely, minimization with respect to $\mathbf{x}$ is done by soft thresholding [6], while the minimization with respect to $\mathbf{f}$ is performed exactly by hard thresholding [2].

Another way to perform the minimization in (1) is to consider another majorization of the objective in (1), motivated by [1]. Let us denote

$$F(\mathbf{x},\,\mathbf{f}) = \left\|\mathbf{u}-\mathbf{D}\mathbf{x}-\mathbf{f}\right\|_2^2. \quad (5)$$

We introduce the surrogate function

$$\begin{aligned}Q_L\left(\mathbf{x},\,\mathbf{f};\,\mathbf{v},\,\mathbf{w}\right) = F(\mathbf{x},\,\mathbf{f}) + (\mathbf{x}-\mathbf{v})^T\,\nabla_{\mathbf{x}}F(\mathbf{v},\,\mathbf{w}) + (\mathbf{f}-\mathbf{w})^T\,\nabla_{f}F(\mathbf{v},\,\mathbf{w}) + &\\ + \frac{L}{2}\left(\left\|\mathbf{x}-\mathbf{v}\right\|_2^2 + \left\|\mathbf{f}-\mathbf{w}\right\|_2^2\right) + \lambda_1\left\|\mathbf{x}\right\|_1. & \quad (6)\end{aligned}$$

Clearly, $Q_L\left(\mathbf{v},\,\mathbf{w};\,\mathbf{v},\,\mathbf{w}\right) = F\left(\mathbf{v},\,\mathbf{w}\right) + \lambda_1\left\|\mathbf{v}\right\|_1$. $Q_L\left(\mathbf{x},\,\mathbf{f};\,\mathbf{v},\,\mathbf{w}\right)$ majorizes $F\left(\mathbf{x},\,\mathbf{f}\right) + \lambda_1\left\|\mathbf{x}\right\|_1$ if L is selected greater than the Lipschitz constant of the gradient of F [1]. We denote

$$p_{L;\,\mathbf{x}}\left(\mathbf{v},\,\mathbf{w}\right) = \underset{\mathbf{x}}{\operatorname{argmin}}\, Q_L\left(\mathbf{x},\,\mathbf{f};\,\mathbf{v},\,\mathbf{w}\right) \quad (7)$$

and

$$p_{L;\,\mathbf{f}}\left(\mathbf{v},\,\mathbf{w}\right) = \underset{\mathbf{f}}{\operatorname{argmin}}\, Q_L\left(\mathbf{x},\,\mathbf{f};\,\mathbf{v},\,\mathbf{w}\right) \quad \text{subject to} \quad \left\|\mathbf{\Omega}\otimes\mathbf{f}\right\|_0 \leq \lambda_2. \quad (8)$$

Now, we have

$$\begin{aligned}F\left(p_{L;\mathbf{x}}(\mathbf{v},\,\mathbf{w}), p_{L;f}(\mathbf{v},\,\mathbf{w})\right) + \left\|p_{L;\mathbf{x}}(\mathbf{v},\,\mathbf{w})\right\|_1 \leq &\\ \leq Q_L\left(p_{L;\mathbf{x}}(\mathbf{v},\,\mathbf{w}), p_{L;\mathbf{f}}(\mathbf{v},\,\mathbf{w});\,\mathbf{v},\,\mathbf{w}\right) \leq F\left(\mathbf{v},\mathbf{w}\right) + \left\|\mathbf{v}\right\|_1. & \quad (9)\end{aligned}$$

Therefore, by iteratively applying $p_{L;\mathbf{x}}$ and $p_{L,\mathbf{f}}$ the objective function is decreased. $p_{L;\mathbf{x}}$ and $p_{L;\mathbf{f}}$ have the form

$$p_{L;\mathbf{x}}\left(\mathbf{v},\,\mathbf{w}\right) = \mathcal{S}_{\frac{\lambda_1}{L}}\left(\mathbf{v} - \frac{1}{L}\mathbf{D}^T\left(\mathbf{D}\,\mathbf{v}+\mathbf{w}-\mathbf{u}\right)\right) \quad (10)$$

where $\mathcal{S}_\lambda$ is the soft thresholding operator defined as

$$\mathcal{S}_\lambda(x) = \begin{cases} x-\lambda & \text{if } x > \lambda \\ 0 & \text{if } |x| \leq \lambda \\ x+\lambda & \text{if } x < -\lambda \end{cases}$$

and

$$p_{L;\mathbf{f}}\left(\mathbf{v},\,\mathbf{w}\right)=\mathcal{H}_{\lambda_2}\left(\mathbf{w}-\frac{1}{L}\left(\mathbf{D}\;\mathbf{v}+\mathbf{w}-\mathbf{u}\right)\right) \tag{11}$$

where $\mathcal{H}_\lambda$ is the hard thresholding operator that sets all but the largest (in absolute value) λ elements of its argument to zero [1].

The above two approaches to solving (1) are equivalent. However, the step size $1/L$ above can generally be too small, which slows down the convergence of the algorithm. Namely, in numerical simulations we found that the choice $L=1$ leads to faster convergence, despite the fact that in our case $L>1$.

2.4 Convergence of the iterative algorithm

The proof of convergence of the proposed algorithm is based on the proof of convergence of the iterative hard thresholding algorithm [2]. General steps of the proof are as follows. First, fixed points of the algorithm are characterized. Then, it is showed that every fixed point of the algorithm is also a local minimum. Finally, we show that the algorithm converges to a point that satisfies necessary and sufficient conditions to be a fixed point. Therefore, the final result is that the algorithm converges to a local minimum of the problem (which is the best we can hope for since the minimization problem is nonconvex).

In the following, we assume that $\|\mathbf{D}\|_2<1$. We start with the lemma characterizing fixed points of the algorithm.

Lemma 1 *Let us denote $\Gamma=\{1,\ldots,n\}$, $\Lambda=\{1,\ldots,m\}$. For $\mathbf{x}\in\mathbb{R}^n$ and $\mathbf{f}\in\mathbb{R}^m$ we also define $\Gamma_1=\operatorname{supp}\mathbf{x}$ and Λ_1 as the indexes of M largest elements of $\mathbf{f}$ (in absolute value). Then, the vector $\left[\mathbf{x}^T\mathbf{f}^T\right]^T$ is a fixed point of the algorithm if and only if*

$$\left(\mathbf{D}^T\left(\mathbf{u}-\mathbf{D}\mathbf{x}-\mathbf{f}\right)\right)_i\begin{cases}=\lambda_1\,\mathrm{sgn}(\mathbf{x}_i) & i\in\Gamma_1\\ \in[-\lambda_1,\,\lambda_1] & i\in\Gamma\setminus\Gamma_1\end{cases} \tag{12}$$

and

$$\left|\left(\mathbf{u}-\mathbf{D}\mathbf{x}-\mathbf{f}\right)_i\right|\begin{cases}=0 & i\in\Lambda_1\\ \le\lambda_M & i\in\Lambda\setminus\Lambda_1\end{cases} \tag{13}$$

where $\lambda_M=\min_{i\in\Lambda_1}\left|\left(\mathbf{u}-\mathbf{D}\mathbf{x}-\mathbf{f}\right)_i\right|$.

The proof is straightforward.

Next we show that every fixed point of the algorithm is also a local minimum. First we need the following lemma.

Lemma 2 *For every $\mathbf{v},\mathbf{h_1}\in\mathbb{R}^n$ and $\mathbf{w},\mathbf{h_2}\in\mathbb{R}^m$ (such that points $p_{L;\mathbf{f}}(\mathbf{v},\mathbf{w})+\mathbf{h_2}$ and $\mathbf{w}$ are feasible, i.e. satisfy the constraint in (1)) we have*

$$Q_L\left(p_{L;\mathbf{x}}(\mathbf{v},\mathbf{w})+\mathbf{h_1},p_{L;\mathbf{f}}(\mathbf{v},\mathbf{w})+\mathbf{h_2};\;\mathbf{v},\mathbf{w}\right)\ge Q_L\left(p_{L;\mathbf{x}}(\mathbf{v},\mathbf{w}),p_{L;\mathbf{f}}(\mathbf{v},\mathbf{w});\;\mathbf{v},\mathbf{w}\right)+\|\mathbf{h_1}\|_2^2+\|\mathbf{h_2}\|_2^2. \tag{14}$$

Proof. For simplicity of notation, we write $\tilde{\mathbf{x}}=\left[\left(p_{L;\mathbf{x}}(\mathbf{v},\mathbf{w})\right)^T\left(p_{L;\mathbf{f}}(\mathbf{v},\mathbf{w})\right)^T\right]^T$, $\tilde{\mathbf{v}}=\left[\mathbf{v}^T\mathbf{w}^T\right]^T$, $\tilde{\mathbf{h}}=\left[\mathbf{h_1}^T\mathbf{h_2}^T\right]^T$ and $\tilde{\mathbf{D}}=[\mathbf{D}\;\mathbf{I_m}]$. The difference $Q_L\left(\tilde{\mathbf{x}}+\tilde{\mathbf{h}};\tilde{\mathbf{v}}\right)-Q_L\left(\tilde{\mathbf{x}};\tilde{\mathbf{v}}\right)$ can be written as

$$\sum_{i=1}^{n+m}L\tilde{\mathbf{h}}_i\left(\tilde{\mathbf{x}}-\tilde{\mathbf{v}}-\frac{1}{L}\tilde{\mathbf{D}}^T\left(\mathbf{y}-\tilde{\mathbf{D}}\tilde{\mathbf{v}}\right)\right)_i\quad+\quad\sum_{j=1}^{n}\lambda_1\left|\tilde{\mathbf{x}}_j+\tilde{\mathbf{h}}_j\right|\quad-\quad\lambda_1\left|\tilde{\mathbf{x}}_j\right|\quad+\quad\frac{L}{2}\left\|\tilde{\mathbf{h}}\right\|_2^2. \tag{15}$$

We have

$$\begin{aligned}\sum_{i=1}^{n}\lambda_1\left|\tilde{\mathbf{x}}_i+\tilde{\mathbf{h}}_i\right|-\lambda_1\left|\tilde{\mathbf{x}}_i\right|+L\tilde{\mathbf{h}}_i\left(\tilde{\mathbf{x}}-\tilde{\mathbf{v}}-\frac{1}{L}\tilde{\mathbf{D}}^T\left(\mathbf{y}-\tilde{\mathbf{D}}\tilde{\mathbf{v}}\right)\right)_i&=\\ =\sum_{i\in\Gamma_1}\lambda_1\left|\tilde{\mathbf{x}}_i+\tilde{\mathbf{h}}_i\right|-\lambda_1\left|\tilde{\mathbf{x}}_i\right|+L\tilde{\mathbf{h}}_i\left(\tilde{\mathbf{x}}-\tilde{\mathbf{v}}-\frac{1}{L}\tilde{\mathbf{D}}^T\left(\mathbf{y}-\tilde{\mathbf{D}}\tilde{\mathbf{v}}\right)\right)_i&+\\ +\sum_{i\in\Gamma\setminus\Gamma_1}\lambda_1\left|\tilde{\mathbf{x}}_i+\tilde{\mathbf{h}}_i\right|-\lambda_1\left|\tilde{\mathbf{x}}_i\right|+L\tilde{h}_i\left(\tilde{\mathbf{x}}-\tilde{\mathbf{v}}-\frac{1}{L}\tilde{\mathbf{D}}^T\left(\mathbf{y}-\tilde{\mathbf{D}}\tilde{\mathbf{v}}\right)\right)_i&\end{aligned} \tag{16}$$

[1] If some elements have the same absolute value, the hard thresholding operator is not uniquely defined, but the set of minimizers is finite, which will be enough for the theoretical analysis.

From

$$p_{L;\mathbf{x}}(\mathbf{v},\mathbf{w}) = \mathcal{S}_{\frac{\lambda_1}{L}}\left(\mathbf{v}+\frac{1}{L}\mathbf{D}^T\left(\mathbf{y}-\tilde{\mathbf{D}}\tilde{\mathbf{v}}\right)\right) \tag{17}$$

we have that, for $i \in \Gamma_1$,

$$p_{L;\mathbf{x}}(\mathbf{v},\mathbf{w}) - \mathbf{v} - \frac{1}{L}\mathbf{D}^T\left(\mathbf{y}-\tilde{\mathbf{D}}\tilde{\mathbf{v}}\right) = -\operatorname{sgn}(p_{L;\mathbf{x}}(\mathbf{v},\mathbf{w}))\ \frac{\lambda_1}{L}$$

(note that $\tilde{\mathbf{x}}_\Gamma = p_{L;\mathbf{x}}(\mathbf{v},\mathbf{w})$). Therefore, the first sum on the right-hand side in (16) is equal to

$$\sum_{i=1}^{n} \lambda_1 \left|\tilde{\mathbf{x}}_i + \tilde{h}_i\right| - \lambda_1 \operatorname{sgn}\left(\tilde{\mathbf{x}}_i\right)\left(\tilde{\mathbf{x}}_i + \tilde{h}_i\right) \geq 0.$$

On the other hand, for $i \in \Gamma \setminus \Gamma_1$

$$\left|\left(\mathbf{D}^T\left(\mathbf{y}-\tilde{\mathbf{D}}\tilde{\mathbf{v}}\right)\right)_i\right| \leq \lambda_1,$$

and therefore, if $\|\mathbf{h_1}\|$ is small enough, the second sum on the right-hand side in (16) is also ≥ 0. We now look at the rest of the sum in (15):

$$\sum_{i=n+1}^{n+m} L\tilde{h}_i\left(\tilde{\mathbf{x}} - \tilde{\mathbf{v}} - \frac{1}{L}\tilde{\mathbf{D}}^T\left(\mathbf{y}-\tilde{\mathbf{D}}\tilde{\mathbf{v}}\right)\right)_i \quad = \quad \sum_{i=1}^{m} L\left(\mathbf{h_2}\right)_i\left(p_{L;f}(\mathbf{v},\mathbf{w}) - \mathbf{w} - \frac{1}{L}\left(\mathbf{y}-\tilde{\mathbf{D}}\tilde{\mathbf{v}}\right)\right)_i \tag{18}$$

Using the notation from Lemma 1, we look separately at the terms in the sum for $i \in \Lambda_1$ and for $i \in \Lambda\setminus\Lambda_1$. For $i \in \Lambda_1$, we have

$$\left(p_{L;f}(\mathbf{v},\mathbf{w})\right)_i = w_i + \frac{1}{L}\left(\mathbf{y}-\tilde{\mathbf{D}}\tilde{\mathbf{v}}\right)_i$$

and therefore the corresponding term in the sum is zero. For $i \notin \operatorname{supp}(p_{L;f}(\mathbf{v},\mathbf{w})) \subseteq \Lambda_1$, we choose $\epsilon > 0$ such that $\min_{i\in\operatorname{supp}(p_{L;f}(\mathbf{v},\mathbf{w}))}\left|\left(p_{L;f}(\mathbf{v},\mathbf{w})\right)_i\right| > \epsilon$. Therefore, since $p_{L;f}(\mathbf{v},\mathbf{w}) + \mathbf{h_2}$ is feasible, $\left(\mathbf{h_2}\right)_i$ has to be zero outside of Λ_1. Therefore, every term in the sum in (18) is zero, and the lemma is proven. □

Lemma 3 *Every fixed point of the algorithm is a local minimum of the problem* (1).

Proof. Let $\left[\left(\mathbf{x}^*\right)^T\left(\mathbf{f}^*\right)^T\right]^T$ be a fixed point of the algorithm. Let $\mathbf{h_1} \in \mathbb{R}^n$ and $\mathbf{h_2} \in \mathbb{R}^m$ such that $\mathbf{f}^* + \mathbf{h_2}$ is feasible and $\|\mathbf{h_1}\|_2$ and $\|\mathbf{h_2}\|_2$ are small enough. We have

$$\begin{aligned}
&\frac{1}{2}\left\|\mathbf{y}-\mathbf{D}\left(\mathbf{x}^*+\mathbf{h_1}\right)-\left(\mathbf{f}^*+\mathbf{h_2}\right)\right\|_2^2 + \lambda_1\left\|\mathbf{x}^*+\mathbf{h_1}\right\|_1 = \\
&\qquad = Q_L\left(\mathbf{x}^*+\mathbf{h_1}, \mathbf{f}^*+\mathbf{h_2}; \mathbf{x}^*, \mathbf{f}^*\right) - \frac{L}{2}\left(\|\mathbf{h_1}\|_2^2 + \|\mathbf{h_2}\|_2^2\right) + \frac{1}{2}\left\|\mathbf{D}\mathbf{h_1}+\mathbf{h_2}\right\|_2^2 \geq \\
&\qquad\qquad \geq Q_L\left(\mathbf{x}^*+\mathbf{h_1}, \mathbf{f}^*+\mathbf{h_2}; \mathbf{x}^*, \mathbf{f}^*\right) - \frac{L}{2}\left(\|\mathbf{h_1}\|_2^2 + \|\mathbf{h_2}\|_2^2\right) \geq \\
&\qquad\qquad\qquad \geq Q_L\left(\mathbf{x}^*, \mathbf{f}^*; \mathbf{x}^*, \mathbf{f}^*\right) = \frac{1}{2}\left\|\mathbf{y}-\mathbf{D}\mathbf{x}^*-\mathbf{f}^*\right\|_2^2 + \lambda_1\left\|\mathbf{x}^*\right\|_1.
\end{aligned} \tag{19}$$

The last inequality follows from Lemma 2. The lemma is proven. □

To finally prove the convergence of the algorithm , we also need the following lemma. It is completely analogous to Lemma D.1 in [2].

Lemma 4 *For every $\epsilon > 0$ there is $k_* \in \mathbb{N}$ such that for all $k > k_*$, $\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|_2^2 + \left\|\mathbf{f}^{(k+1)} - \mathbf{f}^{(k)}\right\|_2^2 \leq \epsilon$.*

Proof. Since $\left\|\tilde{\mathbf{D}}\right\|_2 < L$, where as before $\tilde{\mathbf{D}} = [\mathbf{D}\ \mathbf{I}]$, there is $c > 0$ such that $\mathbf{I} - \frac{1}{L}\tilde{\mathbf{D}}^T\tilde{\mathbf{D}} \succeq c$. Therefore,

$$\sum_{i=0}^{k_*} \left(\left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\|_2^2 + \left\| \mathbf{f}^{(k+1)} - \mathbf{f}^{(k)} \right\|_2^2 \right) \leq$$

$$\leq \frac{1}{c} \sum_{i=0}^{k_*} \left(\left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\|_2^2 + \left\| \mathbf{f}^{(k+1)} - \mathbf{f}^{(k)} \right\|_2^2 - \right.$$

$$\left. - \frac{1}{L} \left\| \mathbf{D} \left(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right) + \left(\mathbf{f}^{(k+1)} - \mathbf{f}^{(k)} \right) \right\|_2^2 \right) \leq$$

$$\leq \frac{2}{cL} \sum_{i=0}^{k_*} \left(F\left(\mathbf{x}^{(k)}, \mathbf{f}^{(k)}\right) - F\left(\mathbf{x}^{(k+1)}, \mathbf{f}^{(k+1)}\right) \right) =$$

$$= \frac{2}{cL} \left(F\left(\mathbf{x}^0, \mathbf{f}^0\right) - F\left(\mathbf{x}^{k_*+1}, \mathbf{f}^{k_*+1}\right) \right) \leq \frac{2}{cL} F\left(\mathbf{x}^0, \mathbf{f}^0\right). \quad (20)$$

Therefore, the series (obtained by letting $k_* \to \infty$) converges and the lemma is proven. □

Finally, we have the following proposition.

Proposition 1 *The algorithm converges to a fixed point, or equivalently, to a local minimum of the problem.*

Proof. Let $\epsilon > 0$ be small enough. According to Lemma 4, there is $k_* > 0$ such that for all $k > k_*$ $\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|_2 \leq \epsilon$ and $\left\|\mathbf{f}^{(k+1)} - \mathbf{f}^{(k)}\right\|_2 \leq \epsilon$. Let us denote $\Gamma_1 = \mathrm{supp}(\mathbf{x}^{(k)})$ and $\Lambda_1 = \mathrm{supp}(\mathbf{f}^{(k)})$. First, we consider the update equation for $\mathbf{f}^{(k+1)}$:

$$\mathbf{f}^{(k+1)} = \mathcal{H}_M \left(\mathbf{f}^{(k)} + \frac{1}{L} \left(\mathbf{y} - \mathbf{D}\mathbf{x}^{(k)} - \mathbf{f}^{(k)} \right) \right). \quad (21)$$

We follow the approach in [2], and consider two cases separately: in the first case, we suppose that there are infinitely many k such that $\mathbf{f}^{(k)}$ and $\mathbf{f}^{(k+1)}$ have different support; in the second case, we suppose that there is $\bar{k}$ such that for all $k > \bar{k}$ the support of $\mathbf{f}^{(k)}$ is fixed.

In the first case, using exactly the same approach as in [2] we show that, for every $\epsilon > 0$ there is $\hat{k} > \max\left(k_*, \bar{k}\right)$ such that, for all $k > \hat{k}$, $\left\|\mathbf{y} - \mathbf{D}\mathbf{x}^{(k)} - \mathbf{f}^{(k)}\right\|_2 \leq \epsilon$. Since any (feasible) point $\mathbf{f}^*$ such that $y - \mathbf{D}\mathbf{x}^* - \mathbf{f}^* = 0$ is a fixed point of the update equation (21), we only need to look at what happens with the sequence $\mathbf{x}^{(k)}$ in this case. Let us recall the update equation for $\mathbf{x}^{(k+1)}$:

$$\mathbf{x}^{(k+1)} = \mathcal{S}_{\frac{\lambda_1}{L}} \left(\mathbf{x}^{(k)} + \frac{1}{L}\mathbf{D}^T \left(\mathbf{y} - \mathbf{D}\mathbf{x}^{(k)} - \mathbf{f}^{(k)} \right) \right). \quad (22)$$

From previous discussion we have (by denoting $r^{(k)} = y - \mathbf{D}\mathbf{x}^{(k)} - \mathbf{f}^{(k)}$) that $\left\|r^{(k)}\right\| \leq \epsilon$ for all large enough k. We also have (for all large enough k) $\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|_2 \leq \epsilon$. Since we can suppose $\epsilon < \frac{\lambda_1}{L}$ (just choose large enough $\hat{k}$, and this will hold for all $k > \hat{k}$), we have $\mathbf{x}^{(k+1)} = 0$, and this holds for all k. However, the point $\mathbf{x}^* = 0$ and $r^* = 0$ is a fixed point of the update equation (22) for $\mathbf{x}^{(k+1)}$. Therefore, in this case the algorithm converges to a fixed point.

In the second case, since the support of $\mathbf{f}^{(k)}$ doesn't change any more for large k, t we can drop the hard thresholding operator from the iteration (21). Therefore, the algorithm reduces to a kind of iterative soft thresholding where ℓ_1 penalty is added only for one part of the vector. Since here we are considering only finite-dimensional setting, the same proof of convergence as presented in [6] is also valid in this case. Namely, in [6] a weighted ℓ_1 norm $\sum_i w_i |x_i|$ in (1) was used, with the assumption $w_i > 0$ for all i. However, Lemma 3.6 in [6] is the only place where the assumption $w_i > 0$, for all i, was used. That lemma states that the sequence of norms $\left\|\left[\left(\mathbf{x}^{(k)}\right)^T \left(\mathbf{f}^{(k)}\right)^T\right]\right\|_2$ is uniformly bounded. However, this is trivially true in the finite-dimensional case, and the assumption $w_i > 0$, for all i, is not necessary. Therefore, the algorithm converges to a fixed point. □

Of course, it is known that the convergence of the iterative soft thresholding algorithm (IST) can be very slow (an improved version of IST was presented in [1]). While it is not obvious how to improve the speed of the proposed algorithm, another possible approach to solving (1) could be to use *alternating* optimization (with respect to $\mathbf{x}$ and $\mathbf{f}$). Recently, a general alternating optimization framework (which applies to formulation (1)) was presented in [3], with established convergence guarantees. However, in our simulations, this approach didn't bring performance improvement.

Table 1: Results of denoising for pure impulse noise in terms of PSNR. Results of the method [22] for images Lena and Barbara are taken as a larger value of the one reported in that paper and the one obtained with our implementation of their method.

		[22]	PARIGI [7]	proposed
Lena	0.4	30.83	**31.75**	30.35
	0.6	26.52	**27.38**	26.88
Bridge	0.4	24.5	**24.8**	24.36
	0.6	21.5	**22.03**	21.5
Baboon	0.4	21.44	**22.02**	21.84
	0.6	**20.23**	20.13	19.75
Barbara	0.4	24.9	**29.92**	24.52
	0.6	21.79	**24.93**	21.95
Boat	0.4	27.25	**27.56**	26.83
	0.6	**24.16**	23.68	23.9
Peppers	0.4	30.1	**31.63**	28.34
	0.6	25.4	**27.58**	26.27
Goldhill	0.4	29.56	**30.06**	29.32
	0.6	25.49	**26.65**	26.3

2.5 Implementation details

As already described, we assume that the dictionary was learned offline. In the first phase of the algorithm, a decreasing sequence of (several) thresholds is selected. For every threshold, a set of possible noisy pixels is selected based on whether their ROAD statistic is above the threshold. Then, an image is processed by discarding these, possibly noisy, pixels. After the whole image is processed, only these noisy pixels are replaced with new values. The process is repeated for all selected thresholds. To reduce computational complexity, since only noisy pixels are replaced in this phase, we process only minimal number of image patches (that cover the entire image). In the second phase, when the influence of noisy pixels is reduced, (1) is solved for overlapping patches to prevent border artifacts. This is the computationally most intensive part of the algorithm. However, it can easily be parallelized to improve the speed of the algorithm if necessary.

3 Experiments

Tables 1, 2 and 3 summarize the results.

For measuring the quality of the reconstructed images, we have used Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity index (SSIM) [21, 20]. PSNR is given by the formula (for 8-bit images)

$$\mathrm{PSNR}(\mathbf{u},\ \tilde{\mathbf{u}}) = -20\log_{10}\frac{\frac{1}{\sqrt{m\cdot n}}\|\mathbf{u}-\tilde{\mathbf{u}}\|_F}{255}$$

where $\mathbf{u}$ denotes the clean image, $\tilde{\mathbf{u}}$ a restored image, and images' size is $m \times n$. It was demonstrated that the SSIM is the metric that better corresponds to subjective quality of visual perception. The SSIM index is computed locally on image patches, within a sliding window that moves pixel-by-pixel across the image; local SSIM measures the similarity of local patch brightness values, contrasts and structures. For more details, we refer the interested reader to the paper [20]. Global SSIM is computed as an average of the SSIM values across the image. It has values between -1 and 1, achieving maximum value 1 if and only if the images being compared are equal. MATLAB code for computing the SSIM index is available at[2].

The methods were tested on test images also used in [7] and provided on their website[3]. For every image, 10 different realizations of random noise were generated, and the results presented in tables 1, 2 and 3 are mean values over these 10 realizations. It should be said that the variations in the results for different noise realizations were small; as described in [7], "standard deviation for all methods is always close to 0.1; therefore, PSNR differences smaller than 0.2 should be considered meaningless in prac- tice".

[2] http://www.ece.uwaterloo.ca/~z70wang/research/ssim/

[3] http://perso.telecom-paristech.fr/~delon/Demos/Impulse/

Table 2: Results of denoising for pure impulse noise in terms of the SSIM measure.

		[22]	PARIGI [7]	proposed
Lena	0.4	0.86	**0.91**	0.88
	0.6	0.73	**0.83**	0.81
Bridge	0.4	0.75	**0.76**	**0.76**
	0.6	0.58	0.55	**0.59**
Baboon	0.4	0.63	0.63	**0.69**
	0.6	**0.5**	0.46	**0.5**
Barbara	0.4	0.75	**0.92**	0.81
	0.6	0.6	**0.82**	0.68
Boat	0.4	0.82	**0.83**	0.82
	0.6	0.67	0.7	**0.71**
Peppers	0.4	0.86	**0.88**	0.86
	0.6	0.7	**0.82**	0.8
Goldhill	0.4	**0.84**	**0.84**	**0.84**
	0.6	0.68	**0.73**	**0.73**

Table 3: Results of denoising for mixed Gaussian-impulse noise in terms of the PSNR measure. For method [22], only the results reported in that paper are included (see text).

		[22]	PARIGI [7]	proposed
Lena	$p = 0.1$, $\sigma = 5$	34.98	34.72	**35**
	$p = 0.3$, $\sigma = 5$	32.04	**32.57**	31.03
	$p = 0.1$, $\sigma = 15$	**30.85**	30.31	30.76
	$p = 0.3$, $\sigma = 15$	29.11	**29.22**	29.15
Bridge	$p = 0.1$, $\sigma = 5$	-	26.96	**28.56**
	$p = 0.3$, $\sigma = 5$	-	**25.45**	25.26
	$p = 0.1$, $\sigma = 15$	-	25.34	**25.82**
	$p = 0.3$, $\sigma = 15$	-	23.38	**23.9**
Baboon	$p = 0.1$, $\sigma = 5$	-	24.81	**25.11**
	$p = 0.3$, $\sigma = 5$	-	**23.05**	22.55
	$p = 0.1$, $\sigma = 15$	-	**23.63**	23.15
	$p = 0.3$, $\sigma = 15$	-	**21.81**	21.58
Barbara	$p = 0.1$, $\sigma = 5$	30.48	**31.55**	28.9
	$p = 0.3$, $\sigma = 5$	25.92	**29.28**	25.42
	$p = 0.1$, $\sigma = 15$	27.31	**28.8**	25.78
	$p = 0.3$, $\sigma = 15$	24.55	**27.33**	24
Cameraman	$p = 0.1$, $\sigma = 5$	-	34.98	**35.63**
	$p = 0.3$, $\sigma = 5$	-	**31.4**	30.08
	$p = 0.1$, $\sigma = 15$	-	30.33	**30.54**
	$p = 0.3$, $\sigma = 15$	-	**28.59**	28.33
Boat	$p = 0.1$, $\sigma = 5$	-	**31.41**	31.13
	$p = 0.3$, $\sigma = 5$	-	**28.81**	27.59
	$p = 0.1$, $\sigma = 15$	-	**28.21**	28.18
	$p = 0.3$, $\sigma = 15$	-	**26.57**	26.3
Peppers	$p = 0.1$, $\sigma = 5$	-	**33.9**	30.11
	$p = 0.3$, $\sigma = 5$	-	**32.38**	28.68
	$p = 0.1$, $\sigma = 15$	-	**30.28**	28.39
	$p = 0.3$, $\sigma = 15$	-	**29.39**	27.63
Goldhill	$p = 0.1$, $\sigma = 5$	-	32.6	**33.1**
	$p = 0.3$, $\sigma = 5$	-	**30.64**	29.85
	$p = 0.1$, $\sigma = 15$	-	29.08	**29.36**
	$p = 0.3$, $\sigma = 15$	-	**27.99**	27.92

Noise parameters (ratio of pure impulse noise and standard deviation of Gaussian noise) were selected as in [7] to enable fair comparison of methods.

Details about the parameters of the proposed method are as follows. In the first phase of the image restoration algorithm, an image is processed 4 times for decreasing sequence of thresholds in the ROAD detector. Namely, parameters of the ROAD detector are: window size is 5×5, the number of largest differences in the window that are used to calculate the ROAD statistic is 12, and the overall threshold above which the pixel is declared as noisy is $12 \cdot \delta$, where the decreasing sequence of δ-s is $[100, 80, 60, 40]$. In every step in the first phase, pixels that are declared as noisy are discarded in the computation, and after the image is processed, only these pixels are replaced. Therefore, after the first phase, many noisy pixels are replaced by approximations of their true values, and the influence of noisy pixels is reduced for the second phase of the algorithm. In the second phase, overlapping image patches are processed. Depending on the noise level, no impulse detection is performed in the second phase, or a high threshold for the ROAD statistic is selected. After the second phase, all image pixels are replaced by the approximations of their true values obtained with the algorithm. The parameters of the iterative $\ell_0 - \ell_1$ minimization algorithm were selected by coarse cross-validation. Therefore, possibly even better results could be obtained by tuning the parameters more carefully. We refer the reader to the MATLAB code of our proposed method, available at author's webpage [4], for exact parameters that were used. All the experiments reported here can easily be reproduced. Parameter μ was set to 1, since this value resulted in faster convergence of the algorithm, despite the (optimal) theoretical value $\mu = 0.5$. The number of iterations was set to 3000. Average time elapsed for the algorithm was about 30 minutes; it should be noted that the algorithm could be parallelized which would make it much faster (the time mentioned above is the result of our "naive" implementation). Tables 1, 2 and 3 summarize the results. For method [22] in the case of mixed Gaussian-impulse, only the results reported in that paper are included, since with our implementation we couldn't find the setting of the parameters that works well.

The method presented here obtained (for some images) comparable results with other two methods for pure impulse noise, especially in terms of the SSIM measure. For mixed Gaussian-impulse noise, the proposed method was generally inferior. The method from [7] is especially superior on Barbara image, as already discussed in [7]. However, proposed method gave good results (in terms of SSIM) for images rich with fine details (Bridge, Baboon, Boat and Goldhill). It should be noted that, in the above experiments, the dictionary was learned on a generic set of natural images, and therefore didn't handle specific textures, like those in Barbara image, well. Better results for specific images could possibly be obtained with more specialized dictionaries or by using several regularizations, as in [23, 15, 18]. It should be noted that none of the referenced papers presented comparisons with recent state-of-the-art method [7]. Also, in most papers (except [7]), only impulse noise ratios below 50 percent were used.

4 Conclusions

We have presented a simple iterative mixed soft-hard thresholding algorithm for solving inverse problems with ℓ_0-ℓ_1 sparsity constraints, and its application to image restoration under impulse noise. The theoretical analysis of the algorithm is simple since it is based on the analysis of iterative soft and hard thresholding algorithms. Image regularization used in this paper is based on sparse representations in learned dictionary. Independent component analysis (ICA) was used for dictionary learning purpose. Namely, it seems, based on the presented experiments, that the dictionary learned using ICA tends to give (at least slightly) better results in terms of the Structural SIMilarity (SSIM) index. Although other approaches discussed in the paper perform much better for some images and some problem settings, the approach proposed here performs at least slightly better for some images. This agrees with the statement from [7], namely that the performance of a denoising method depends on the image (or texture) class. We also note that the algorithm presented here could possibly be applied to general inverse problems, for example robust regression.

References

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[4] http://www.lair.irb.hr/ikopriva/marko-filipovi.html

[2] Thomas Blumensath and Mike E. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14(5-6):629–654, 2008.

[3] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, pages 1–36, 2013.

[4] Jian-Feng Cai, Raymond H. Chan, and Mila Nikolova. Two-phase approach for deblurring images corrupted by impulse plus gaussian noise. *Inverse Problems and Imaging*, 2(2):187–204, 2008.

[5] Jian-Feng Cai, Raymond H. Chan, and Mila Nikolova. Fast two-phase image deblurring under impulse noise. *Journal of Mathematical Imaging and Vision*, 36(1):46–53, 2010.

[6] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[7] J. Delon and A. Desolneux. A patch-based approach for removing impulse or mixed gaussian-impulse noise. *SIAM Journal on Imaging Sciences*, 6(2):1140–1174, 2013.

[8] Yiqiu Dong, R.H. Chan, and Shufang Xu. A detection statistic for random-valued impulse noise. *Image Processing, IEEE Transactions on*, 16(4):1112–1120, 2007.

[9] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006.

[10] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *Image Processing, IEEE Transactions on*, 17(10):1737–1754, 2008.

[11] R. Garnett, T. Huegerich, C. Chui, and Wenjie He. A universal noise removal algorithm with an impulse detector. *Image Processing, IEEE Transactions on*, 14(11):1747–1754, 2005.

[12] Weihong Guo, Jing Qin, and Wotao Yin. A new detail-preserving regularity scheme. Technical report, 2013.

[13] A. Jezierska, E. Chouzenoux, J.-C. Pesquet, and H. Talbot. A convex approach for image restoration with exact poisson-gaussian likelihood. Technical report, 2013.

[14] Yan-Ran Li, Lixin Shen, Dao-Qing Dai, and B.W. Suter. Framelet algorithms for de-blurring images corrupted by impulse plus gaussian noise. *Image Processing, IEEE Transactions on*, 20(7):1822–1837, 2011.

[15] L. Ma, J. Yu, and T. Zeng. Sparse representation prior and total variation–based image deblurring under impulse noise. *SIAM Journal on Imaging Sciences*, 6(4):2258–2284, 2013.

[16] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *Image Processing, IEEE Transactions on*, 17(1):53–69, 2008.

[17] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, March 2010.

[18] Gabriel Peyré, Jalal Fadili, and Jean-Luc Starck. Learning adapted dictionaries for geometry and texture separation, 2007.

[19] Y. Wang, A. Szlam, and G. Lerman. Robust locally linear analysis with applications to image denoising and blind inpainting. *SIAM Journal on Imaging Sciences*, 6(1):526–562, 2013.

[20] Zhou Wang and A.C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *Signal Processing Magazine, IEEE*, 26(1):98–117, 2009.

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

[22] Y. Xiao, T. Zeng, J. Yu, and M. K. Ng. Restoration of images corrupted by mixed gaussian-impulse noise via l1-l0 minimization. *Pattern Recognition*, 44(8):1708 – 1720, 2011.

[23] M. Yan. Restoration of images corrupted by impulse noise and mixed gaussian impulse noise using blind inpainting. *SIAM Journal on Imaging Sciences*, 6(3):1227–1245, 2013.

[24] Guoshen Yu, G. Sapiro, and S. Mallat. Image modeling and enhancement via structured sparse model selection. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1641–1644, 2010.