ℓ_0TV: A New Method for Image Restoration in the Presence of Impulse Noise

Ganzhao Yuan[1] and Bernard Ghanem[2]
[1]South China University of Technology (SCUT), P.R. China
[2]King Abdullah University of Science and Technology (KAUST), Saudi Arabia
yuanganzhao@gmail.com, bernard.ghanem@kaust.edu.sa

Abstract

Total Variation (TV) is an effective and popular prior model in the field of regularization-based image processing. This paper focuses on TV for image restoration in the presence of impulse noise. This type of noise frequently arises in data acquisition and transmission due to many reasons, e.g. a faulty sensor or analog-to-digital converter errors. Removing this noise is an important task in image restoration. State-of-the-art methods such as Adaptive Outlier Pursuit(AOP) [42], which is based on TV with ℓ_{02}-norm data fidelity, only give sub-optimal performance. In this paper, we propose a new method, called ℓ_0TV-PADMM, which solves the TV-based restoration problem with ℓ_0-norm data fidelity. To effectively deal with the resulting non-convex non-smooth optimization problem, we first reformulate it as an equivalent MPEC (Mathematical Program with Equilibrium Constraints), and then solve it using a proximal Alternating Direction Method of Multipliers (PADMM). Our ℓ_0TV-PADMM method finds a desirable solution to the original ℓ_0-norm optimization problem and is proven to be convergent under mild conditions. We apply ℓ_0TV-PADMM to the problems of image denoising and deblurring in the presence of impulse noise. Our extensive experiments demonstrate that ℓ_0TV-PADMM outperforms state-of-the-art image restoration methods.

1. Introduction

Image restoration is an inverse problem, which aims at estimating the original *clean* image $\mathbf{u}$ from a blurry and/or noisy observation $\mathbf{b}$. Mathematically, this problem is formulated as:

$$\mathbf{b} = (\mathbf{K}\mathbf{u} \odot \boldsymbol{\varepsilon}_m) + \boldsymbol{\varepsilon}_a, \tag{1}$$

where $\mathbf{K}$ is a linear operator, $\boldsymbol{\varepsilon}_m$ and $\boldsymbol{\varepsilon}_a$ are the noise vectors, and $\odot$ denotes an elementwise product. Let $\mathbf{1}$ and $\mathbf{0}$ be column vectors of all entries equal to one and zero, respectively. When $\boldsymbol{\varepsilon}_m = \mathbf{1}$ and $\boldsymbol{\varepsilon}_a \neq \mathbf{0}$ (or $\boldsymbol{\varepsilon}_m \neq \mathbf{0}$ and $\boldsymbol{\varepsilon}_a = \mathbf{0}$), Eq (1) corresponds to the additive (or multiplicative) noise model. For convenience, we adopt the vector representation for images, where a 2D $M \times N$ image is column-wise stacked into a vector $\mathbf{u} \in \mathbb{R}^{M \times N}$. So, for completeness, we have $\mathbf{1}, \mathbf{0}, \mathbf{b}, \mathbf{u}, \boldsymbol{\varepsilon}_a, \boldsymbol{\varepsilon}_m \in \mathbb{R}^n$, and $\mathbf{K} \in \mathbb{R}^{n \times n}$.

In general image restoration problems, $\mathbf{K}$ represents a certain linear operator, e.g. convolution, wavelet transform, etc., and recovering $\mathbf{u}$ from $\mathbf{b}$ is known as image deconvolution or image deblurring. When $\mathbf{K}$ is the identity operator, estimating $\mathbf{u}$ from $\mathbf{b}$ is referred to as image denoising [35]. The problem of estimating $\mathbf{u}$ from $\mathbf{b}$ is called a linear inverse problem which, for most scenarios of practical interest, is ill-posed due to the singularity and/or the ill-conditioning of $\mathbf{K}$. Therefore, in order to stabilize the recovery of $\mathbf{u}$, it is necessary to incorporate prior-enforcing regularization on the solution. Therefore, image restoration can be modelled globally as the following optimization problem:

$$\min_{\mathbf{u}} \; \ell(\mathbf{K}\mathbf{u}, \mathbf{b}) + \lambda \, \Omega(\boldsymbol{\nabla}_x \mathbf{u}, \boldsymbol{\nabla}_y \mathbf{u}) \tag{2}$$

where $\ell(\mathbf{K}\mathbf{u}, \mathbf{b})$ measures the data fidelity between $\mathbf{K}\mathbf{u}$ and the observation $\mathbf{b}$ and $\boldsymbol{\nabla}_x \in \mathbb{R}^{n \times n}$ and $\boldsymbol{\nabla}_y \in \mathbb{R}^{n \times n}$ are two suitable linear transformation matrices such that $\boldsymbol{\nabla}_x \mathbf{u} \in \mathbb{R}^n$ and $\boldsymbol{\nabla}_y \mathbf{u} \in \mathbb{R}^n$ compute the discrete gradients of the image u along the x-axis and y-axis, respectively[1], $\Omega(\boldsymbol{\nabla}_x \mathbf{u}, \boldsymbol{\nabla}_y \mathbf{u})$ is the regularizer on $\boldsymbol{\nabla}_x \mathbf{u}$ and $\boldsymbol{\nabla}_y \mathbf{u}$, and λ is a

[1]In practice, one does not need to compute and store the matrices $\boldsymbol{\nabla}_x$ and $\boldsymbol{\nabla}_y$ explicitly. Since the adjoint of the gradient operator $\boldsymbol{\nabla}$ is the negative divergence operator $-\mathrm{div}$, i.e., $\langle \mathbf{r}, \boldsymbol{\nabla}_x \mathbf{u} \rangle = \langle -\mathrm{div}_x \mathbf{r}, \mathbf{u} \rangle, \langle \mathbf{s}, \boldsymbol{\nabla}_y \mathbf{u} \rangle = \langle -\mathrm{div}_y \mathbf{s}, \mathbf{u} \rangle$ for any

positive parameter used to balance the two terms for minimization. Apart from regularization, other prior information such as bound constraints [4, 46] or hard constraints can be incorporated into the general optimization framework in Eq (2).

1.1. Related Work

This subsection presents a brief review of existing TV methods, from the viewpoint of regularization, data fidelity, and optimization algorithms. For more discussions on the connection with existing work, please refer to the **supplementary material**.

Regularization: Several regularization models have been studied in the literature (see Table 1). The Tikhonov-like regularization [1] function Ω_{tik} is quadratic and smooth, therefore it is relatively inexpensive to minimize with first-order smooth optimization methods. However, since this method tends to overly smooth images, it often erodes strong edges and texture details. To address this issue, the total variation (TV) regularizer was proposed by Rudin, Osher and Fatemi in [33] for image denoising. Several other variants of TV have been extensively studied. The original TV norm Ω_{tv_2} in [33] is isotropic, while an anisotropic variation Ω_{tv_1} is also used. From a numerical point of view, Ω_{tv_2} and Ω_{tv_1} cannot be directly minimized since they are not differentiable. A popular method is to use their smooth approximation Ω_{stv} and Ω_{hub} (see [32] for details). Very recently, the Potts model [19, 28] Ω_{pot}, which is based on the ℓ_0-norm, has received much attention. It has been shown to be particularly effective for image smoothing [40] and motion deblurring [41]. For more applications of the Potts model, we refer the reader to [5, 9].

Data Fidelity Models: The fidelity function $\ell(\cdot,\cdot)$ in Eq (2) usually penalizes the difference between $\mathbf{Ku}$ and $\mathbf{b}$ by using different norms/divergences. Its form depends on the assumed distribution of the noise model. Some typical noise models and their corresponding fidelity terms are listed in Table 2. The classical TV model [33] only considers TV minimization involving the squared ℓ_2 fidelity term for recovering images corrupted by additive Gaussian noise. However, this model is far from optimal when the noise is not Gaussian. Other works [43, 16] extend classical TV to use the ℓ_1-norm in the fidelity term. This norm is suitable for image restoration in the presence of Laplace noise. Moreover, additive uniform noise [15, 36], multiplicative Poisson noise [26], and multiplicative Gamma noise [39] have been considered in the literature. Recently, a sparse noise model using an ℓ_{02}-norm for data fidelity has been investigated in [42] to remove impulse and mixed Gaussian impulse noise. In this paper, we consider ℓ_0-norm data fidelity and show that it is particularly suitable for reconstructing images corrupted with impulse noise.

Table 1: Regularization Models

Regularization Function	Desc. and Ref.
$\Omega_{\mathrm{tik}}(\mathbf{g},\mathbf{h}) = \sum_{i=1}^{n} \mathbf{g}_i^2 + \mathbf{h}_i^2$	Tikhonov-like, [1]
$\Omega_{\mathrm{tv}_2}(\mathbf{g},\mathbf{h}) = \sum_{i=1}^{n} (\mathbf{g}_i^2 + \mathbf{h}_i^2)^{\frac{1}{2}}$	isotropic, [33, 39]
$\Omega_{\mathrm{tv}_1}(\mathbf{g},\mathbf{h}) = \sum_{i=1}^{n} \lvert\mathbf{g}_i\rvert + \lvert\mathbf{h}_i\rvert$	anisotropic, [35, 43]
$\Omega_{\mathrm{stv}}(\mathbf{g},\mathbf{h}) = \sum_{i=1}^{n} (\mathbf{g}_i^2 + \mathbf{h}_i^2 + \varepsilon^2)^{\frac{1}{2}}$	smooth TV, [12, 36]
$\Omega_{\mathrm{hub}}(\mathbf{g},\mathbf{h}) = \sum_{i=1}^{n} \varphi(\mathbf{g}_i;\mathbf{h}_i)$ $\varphi(\mathbf{g}_i;\mathbf{h}_i) = \begin{cases} \frac{\varepsilon}{2}\lVert\mathbf{g}_i;\mathbf{h}_i\rVert_2^2; & \lVert\mathbf{g}_i;\mathbf{h}_i\rVert_2 \leq \frac{1}{\varepsilon} \\ \lVert\mathbf{g}_i;\mathbf{h}_i\rVert_2 - \frac{1}{2\varepsilon}; & \text{otherwise} \end{cases}$	Huber-like, [32]
$\Omega_{\mathrm{pot}}(\mathbf{g},\mathbf{h}) = \sum_{i=1}^{n} \lvert\mathbf{g}_i\rvert_0 + \lvert\mathbf{h}_i\rvert_0$	Potts model, [40, 41]

Table 2: Data Fidelity Models

Data Fidelity Function	Noise and Ref.
$\ell_2(\mathbf{Ku},\mathbf{b}) = \lVert\mathbf{Ku}-\mathbf{b}\rVert_2^2$	add. Gaussian [33, 8]
$\ell_1(\mathbf{Ku},\mathbf{b}) = \lVert\mathbf{Ku}-\mathbf{b}\rVert_1$	add. Laplace [43, 16]
$\ell_\infty(\mathbf{Ku},\mathbf{b}) = \lVert\mathbf{Ku}-\mathbf{b}\rVert_\infty$	add. uniform [15, 36]
$\ell_p(\mathbf{Ku},\mathbf{b}) = \langle\mathbf{Ku}-\mathbf{b}\odot\log(\mathbf{Ku}),\mathbf{1}\rangle$	mult. Poisson [26, 34]
$\ell_g(\mathbf{Ku},\mathbf{b}) = \langle\log(\mathbf{Ku})+\mathbf{b}\odot\frac{1}{\mathbf{Ku}},\mathbf{1}\rangle$	mult. Gamma [2, 39]
$\ell_{02}(\mathbf{Ku},\mathbf{b}) = \lVert\mathbf{Ku}-\mathbf{b}+\mathbf{z}\rVert_2^2$ $s.t.\ \lVert\mathbf{z}\rVert_0 \leq k$	mixed Gaussian impulse [42, 45]
$\ell_0(\mathbf{Ku},\mathbf{b}) = \lVert\mathbf{Ku}-\mathbf{b}\rVert_0$	add./mult. impulse [13],**[this paper]**

Optimization Algorithms: The optimization problems involved in TV-based image restoration are usually difficult due to the non-differentiability of the TV norm and the high dimensionality of the image data. In the past several decades, a plethora of approaches have been proposed, which include time-marching PDE methods based on the Euler-Lagrange equation [33], the interior-point method [12], the semi-smooth Newton method [31], the second-order cone optimization method [21], the splitting Bregman method [22, 44], the fixed-point iterative method [14], Nesterov's first-order optimal method [30, 4, 37], and alternating direction methods [35, 23, 39]. Among these methods, some solve the TV problem in its primal form [35], while others consider its dual or primal-dual forms [12, 16].

$\mathbf{r},\mathbf{s} \in \mathbb{R}^n$, the inner product between vectors can be evaluated efficiently. Fore more details on the computation of $\boldsymbol{\nabla}$ and div operators, please refer to [8, 36, 3].

In this paper, we handle the TV problem with ℓ_0-norm data fidelity using a primal-dual formulation, where the resulting equality constrained optimization is solved using proximal Alternating Direction Methods of Multipliers (PADMM). It is worthwhile to note that the Penalty Decomposition Algorithm (PDA) in [27] can also solve our problem, however, it lacks numerical stability. The penalty function can be very large ($\geq 10^8$), and the solution can be degenerate when the minimization subproblem is not solved exactly. This motivates us to design a new ℓ_0-norm optimization algorithm in this paper.

1.2. Contributions and Organization

The main contributions of this paper are two-fold. **(1)** ℓ_0-norm data fidelity is proposed to address the TV-based image restoration problem. Compared with existing models, our model is particularly suitable for image restoration in the presence of impulse noise. **(2)** To deal with the resulting NP-hard[2] ℓ_0 norm optimization, we propose a proximal ADMM to solve an equivalent MPEC form of the problem.

The rest of the paper is organized as follows. Section 2 presents the motivation and formulation of the problem for impulse noise removal. Section 3 presents the equivalent MPEC problem and our proximal ADMM solution. Section 4 provides extensive and comparative results in favor of our ℓ_0TV method. Finally, Section 5 concludes the paper.

2. Motivation and Formulations

2.1. Motivation

This work focuses on image restoration in the presence of impulse noise, which is very common in data acquisition and transmission due to faulty sensors or analog-to-digital converter errors, etc. Moreover, scratches in photos and video sequences can be also viewed as a special type of impulse noise. However, removing this kind of noise is not easy, since corrupted pixels are randomly distributed in the image and the intensities at corrupted pixels are usually indistinguishable from those of their neighbors. There are two main types of impulse noise in the literature [16, 25]: random-valued and salt-and-pepper impulse noise. Let $[u_{\min}, u_{\max}]$ be the dynamic range of an image, where $u_{\min} = 0$ and $u_{\max} = 1$ in this paper. We also denote the original and corrupted intensity values at position i as $\mathbf{u}_i$ and $\mathcal{T}(\mathbf{u}_i)$, respectively.

[2]The ℓ_0 norm problem is known to be NP-hard [29], since it is equivalent to NP-complete subset selection problems.

Random-valued impulse noise: A certain percentage of pixels are altered to take on a uniform random number $d_i \in [u_{\min}, u_{\max}]$.

$$\mathcal{T}(\mathbf{u}_i) = \begin{cases} d_i, & \text{with probability } r_{rv} \\ (\mathbf{K}\mathbf{u})_i, & \text{with probability } 1 - r_{rv} \end{cases} \quad (3)$$

Salt-and-pepper impulse noise: A certain percentage of pixels are altered to be either $u_{\min}$ or $u_{\max}$.

$$\mathcal{T}(\mathbf{u}_i) = \begin{cases} u_{\min}, & \text{with probability } r_{sp}/2 \\ u_{\max}, & \text{with probability } r_{sp}/2 \\ (\mathbf{K}\mathbf{u})_i, & \text{with probability } 1 - r_{sp} \end{cases} \quad (4)$$

The above definition means that impulse noise corrupts a portion of pixels in the image while keeping other pixels unaffected. Expectation maximization could be used to find the MAP estimate of $\mathbf{u}$ by maximizing the conditional posterior probability $p(\mathbf{u}|\mathcal{T}(\mathbf{u}))$, the probability that $\mathbf{u}$ occurs when $\mathcal{T}(\mathbf{u})$ is observed. The MAP estimate of $\mathbf{u}$ can be obtained by solving the following optimization problem.

$$\max_{\mathbf{u}} \; \log p(\mathcal{T}(\mathbf{u})|\mathbf{u}) + \log p(\mathbf{u}). \quad (5)$$

We now focus on the two terms in Eq (5). (i) The expression $p(\mathcal{T}(\mathbf{u})|\mathbf{u})$ can be viewed as a fidelity term measuring the discrepancy between the estimate $\mathbf{u}$ and the noisy image $\mathcal{T}(\mathbf{u})$. The choice of the likelihood $p(\mathcal{T}(\mathbf{u})|\mathbf{u})$ depends upon the property of noise. From the definition of impulse noise given above, we have that

$$p(\mathcal{T}(\mathbf{u})|\mathbf{u}) \;\;=\;\; 1 - r = \frac{n - \|\mathcal{T}(\mathbf{u}) - \mathbf{b}\|_0}{n},$$

where r is the noise density level as defined in Eq (3) and Eq (4) and $\|\cdot\|_0$ counts the number of nonzero elements in a vector. (ii) The term $p(\mathbf{u})$ in Eq (5) is used to regularize a solution that has a low probability. We use a TV prior of the form: $p(\mathbf{u}) = \frac{1}{\vartheta}\exp(-\sigma \cdot \Omega_{\text{tv}}(\boldsymbol{\nabla}_x \mathbf{u}, \nabla_y \mathbf{u}))$, where ϑ is a normalization factor, $\sigma\ \Omega_{\text{tv}}(\boldsymbol{\nabla}_x \mathbf{u}, \boldsymbol{\nabla}_y \mathbf{u})$ is the TV prior. Replacing $p(\mathcal{T}(\mathbf{u})|\mathbf{u})$ and $p(\mathbf{u})$ into Eq (5) and ignoring a constant, we obtain the following $\ell_0 TV$ model:

$$\min_{\mathbf{u}} \; \|\mathbf{K}\mathbf{u} - \mathbf{b}\|_0 + \lambda \sum_{i=1}^{n} \Big[|(\nabla_x \mathbf{u})_i|^p + |(\nabla_y \mathbf{u})_i|^p \Big]^{1/p},$$

where λ is a positive number related to ϑ, σ, and r. The parameter p can be 1 (isotropic TV) or 2 (anisotropic TV), and $(\boldsymbol{\nabla}_x \mathbf{u})_i$ and $(\nabla_y \mathbf{u})_i$ denote the ith component of the vectors $\boldsymbol{\nabla}_x \mathbf{u}$ and $\boldsymbol{\nabla}_y \mathbf{u}$, respectively. For convenience, we define $\forall \mathbf{x} \in \mathbb{R}^{2n}$:

$$\|\mathbf{x}\|_{p,1} \triangleq \sum_{i=1}^{n} (|\mathbf{x}_i|^p + |\mathbf{x}_{n+i}|^p)^{\frac{1}{p}}; \;\; \boldsymbol{\nabla} \triangleq \begin{bmatrix} \boldsymbol{\nabla}_x \\ \nabla_y \end{bmatrix} \in \mathbb{R}^{2n \times n}.$$

In order to make use of more prior information, we consider the following box-constrained model:

$$\min_{\mathbf{0} \leq \mathbf{u} \leq \mathbf{1}} \; \|\mathbf{o} \odot (\mathbf{K}\mathbf{u} - \mathbf{b})\|_0 + \lambda \|\boldsymbol{\nabla} \mathbf{u}\|_{p,1} \tag{6}$$

where $\mathbf{o} \in \{0, 1\}$ is specified by the user. For example, in our experiments, we set $\mathbf{o} = \mathbf{1}$ for the random-valued impulse noise and $\mathbf{o}_i = \begin{cases} 0, & \mathbf{b}_i = u_{\min} \text{ or } u_{\max} \\ 1, & \text{otherwise} \end{cases}$ for the salt-and-pepper impulse noise.

In what follows, we focus on optimizing the general formulation in Eq (6). But first, we present an image restoration example on the well-known 'barbara' image using our proposed $\ell_0 TV$-PADMM method for solving Eq (6) in Figure 1.

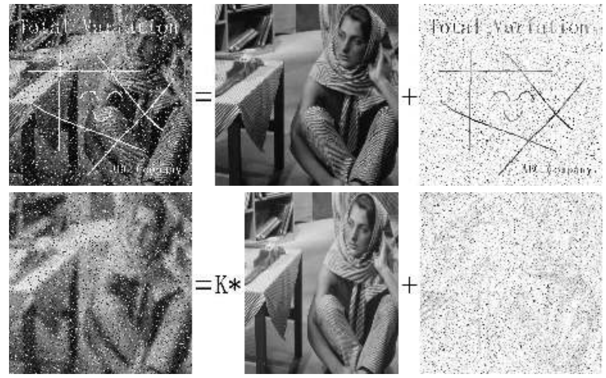

Figure 1: An example of an image recovery result using our proposed ℓ_0TV-PADMM method. Left column: corrupted image. Middle column: recovered image. Right column: absolute residual between these two images.

2.2. Equivalent MPEC Reformulations

In this section, we reformulate the problem in Eq (6) as an equivalent MPEC from a primal-dual viewpoint. First, we provide the variational characterization of the ℓ_0-norm using the following lemma.

Lemma 1. *For any given* $\mathbf{w} \in \mathbb{R}^n$*, it holds that*

$$\|\mathbf{w}\|_0 = \min_{\mathbf{0} \leq \mathbf{v} \leq \mathbf{1}} \; \langle \mathbf{1}, \mathbf{1} - \mathbf{v} \rangle, \;\; s.t. \; \mathbf{v} \odot |\mathbf{w}| = \mathbf{0}, \tag{7}$$

and $\mathbf{v}^* = \mathbf{1} - \text{sign}(|\mathbf{w}|)$ *is the unique optimal solution of the minimization problem in Eq(7).*

Proof. Refer to the **supplementary material**. □

The result of Lemma 1 implies that the ℓ_0-norm minimization problem in Eq(6) is equivalent to

$$\begin{aligned} \min_{\mathbf{0} \leq \mathbf{u}, \mathbf{v} \leq \mathbf{1}} \; & \langle \mathbf{1}, \mathbf{1} - \mathbf{v} \rangle + \lambda \|\boldsymbol{\nabla} \mathbf{u}\|_{p,1} \\ \text{s.t.} \quad & \mathbf{v} \odot |\mathbf{o} \odot (\mathbf{K}\mathbf{u} - \mathbf{b})| = \mathbf{0} \end{aligned} \tag{8}$$

If $\mathbf{u}^*$ is a global optimal solution of Eq (6), then $(\mathbf{u}^*, \mathbf{1} - \text{sign}(|\mathbf{K}\mathbf{u}^* - \mathbf{b}|))$ is globally optimal to Eq (8). Conversely, if $(\mathbf{u}^*, \mathbf{v}^*)$ is a global optimal solution of Eq (8), then $\mathbf{u}^*$ is globally optimal to Eq (6).

Although the MPEC problem in Eq (8) is obtained by increasing the dimension of the original ℓ_0-norm problem in Eq (6), this does not lead to additional local optimal solutions. Moreover, compared with Eq (6), Eq (8) is a non-smooth non-convex minimization problem and its non-convexity is only caused by the complementarity constraint $\mathbf{v} \odot |\mathbf{o} \odot (\mathbf{K}\mathbf{u} - \mathbf{b})| = \mathbf{0}$.

Such a variational characterization of the ℓ_0-norm is proposed in [17, 24, 18], but it is not used to develop any optimization algorithms for ℓ_0-norm problems. We argue that, from a practical perspective, improved solutions to Eq (6) can be obtained by reformulating the ℓ_0-norm in terms of complementarity constraints. In the following section, we will develop an algorithm to solve Eq (8) based on proximal ADMM and show that such a "lifting" technique can achieve a desirable solution of the original ℓ_0-norm optimization problem.

3. Proposed Optimization Algorithm

This section is devoted to the solution of Eq (8). This problem is rather difficult to solve, because it is neither convex nor smooth. Our solution is based on the proximal ADMM method, which iteratively updates the primal and dual variables of the augmented Lagrangian function of Eq (8).

First, we introduce two auxiliary vectors $\mathbf{x} \in \mathbb{R}^{2n}$ and $\mathbf{y} \in \mathbb{R}^n$ to reformulate Eq (8) as:

$$\begin{aligned} & \min\nolimits_{\mathbf{0} \leq \mathbf{u}, \mathbf{v} \leq \mathbf{1}} \; \langle \mathbf{1}, \mathbf{1} - \mathbf{v} \rangle + \lambda \|\mathbf{x}\|_{p,1} \\ & \text{s.t.} \quad \boldsymbol{\nabla}\mathbf{u} = \mathbf{x}, \;\; \mathbf{K}\mathbf{u} - \mathbf{b} = \mathbf{y}, \;\; \mathbf{v} \odot \mathbf{o} \odot |\mathbf{y}| = \mathbf{0} \end{aligned} \tag{9}$$

Let $\mathcal{L}_\beta : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{2n} \times \mathbb{R}^n \times \mathbb{R}^{2n} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be the augmented Lagrangian function of Eq (9).

$$\begin{aligned} & \mathcal{L}_\beta(\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y}, \boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{\pi}) := \langle \mathbf{1}, \mathbf{1} - \mathbf{v} \rangle + \lambda \|\mathbf{x}\|_{p,1} + \\ & \langle \boldsymbol{\nabla}\mathbf{u} - \mathbf{x}, \boldsymbol{\xi} \rangle + \frac{\beta}{2} \|\boldsymbol{\nabla}\mathbf{u} - \mathbf{x}\|^2 + \langle \mathbf{K}\mathbf{u} - \mathbf{b} - \mathbf{y}, \boldsymbol{\zeta} \rangle + \\ & \frac{\beta}{2} \|\mathbf{K}\mathbf{u} - \mathbf{b} - \mathbf{y}\|^2 + \langle \mathbf{v} \odot \mathbf{o} \odot |\mathbf{y}|, \boldsymbol{\pi} \rangle + \frac{\beta}{2} \|\mathbf{v} \odot \mathbf{o} \odot |\mathbf{y}|\|^2, \end{aligned}$$

where $\boldsymbol{\xi}$, $\boldsymbol{\zeta}$ and $\boldsymbol{\pi}$ are the Lagrange multipliers associated with the constraints $\nabla\mathbf{u} = \mathbf{x}$, $\mathbf{Ku} - \mathbf{b} = \mathbf{y}$ and $\mathbf{v} \odot \mathbf{o} \odot |\mathbf{y}| = 0$, respectively, and $\beta > 0$ is the penalty parameter. The detailed iteration steps of the proximal ADMM for Eq (9) are described in Algorithm 1. In simple terms, ADMM updates are performed by optimizing for a set of primal variables at a time, while keeping all other primal and dual variables fixed. The dual variables are updated by gradient ascent on the resulting dual problem. In Algorithm 1, for convenience, we denote the augmented Lagrange function at the k^{th} iteration as $\mathcal{L}_\beta^k(\cdot)$, where all the primal and dual variables except the indicated function argument(s) are fixed to their current estimates.

Algorithm 1 Proximal ADMM (PADMM) for the Non-Convex MPEC in Eq (9)

(S.0) Choose a starting point $(\mathbf{u}^0, \mathbf{v}^0, \mathbf{x}^0, \mathbf{y}^0, \boldsymbol{\xi}^0, \boldsymbol{\zeta}^0)$. Set $k = 0$. Select the parameters $\beta = 1$ and $\kappa \in \left(0, \frac{1}{\beta\|\nabla\|^2 + \beta\|\mathbf{K}\|^2}\right)$.

(S.1) Solve the following minimization problems with $\mathbf{D} := \frac{1}{\kappa}\mathbf{I} - (\beta\nabla^T\nabla + \beta\mathbf{K}^T\mathbf{K})$:

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{0}\leq\mathbf{u}\leq\mathbf{1}} \ \mathcal{L}_\beta^k(\mathbf{u}) + \frac{1}{2}\|\mathbf{u} - \mathbf{u}^k\|_{\mathbf{D}}^2 \quad (10)$$

$$\mathbf{v}^{k+1} = \arg\min_{\mathbf{0}\leq\mathbf{v}\leq\mathbf{1}} \ \mathcal{L}_\beta^k(\mathbf{v}) \quad (11)$$

$$(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}) = \arg\min_{\mathbf{x},\mathbf{y}} \ \mathcal{L}_\beta^k(\mathbf{x}, \mathbf{y}). \quad (12)$$

(S.2) Update the Lagrange multipliers:

$$\boldsymbol{\xi}^{k+1} = \boldsymbol{\xi}^k + \beta(\nabla\mathbf{u}^k - \mathbf{x}^k), \quad (13)$$

$$\boldsymbol{\zeta}^{k+1} = \boldsymbol{\zeta}^k + \beta(\mathbf{Ku}^k - \mathbf{b} - \mathbf{y}^k), \quad (14)$$

$$\boldsymbol{\pi}^{k+1} = \boldsymbol{\pi}^k + \beta(\mathbf{o} \odot \mathbf{v}^k \odot |\mathbf{y}^k|). \quad (15)$$

(S.3) if (k is a multiple of 30), then $\beta = \beta \times \sqrt{10}$

(S.4) Set $k := k + 1$ and then go to Step (S.1).

Next, we focus our attention on the solutions of subproblems (10-12) arising in Algorithm 1.

(i) $\mathbf{u}$-subproblem. Proximal ADMM introduces a convex proximal term $\frac{1}{2}\|\mathbf{u} - \mathbf{u}^k\|_{\mathbf{D}}^2$ to the objective, which leads to a strong convex minimization

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{0}\leq\mathbf{u}\leq\mathbf{1}} \ \frac{\beta}{2}\|\nabla\mathbf{u} - \mathbf{x}^k + \boldsymbol{\xi}^k/\beta\|^2 + \frac{\beta}{2}\|\mathbf{Ku} - \mathbf{b} - \mathbf{y}^k + \boldsymbol{\zeta}^k/\beta\|^2 + \frac{1}{2}\|\mathbf{u} - \mathbf{u}^k\|_{\mathbf{D}}^2. \quad (16)$$

After an elementary calculation, subproblem (16) can be simplified as

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{0}\leq\mathbf{u}\leq\mathbf{1}} \frac{1}{2\kappa}\|\mathbf{u} - \mathbf{g}^k\|^2$$

with $\mathbf{g}^k = \mathbf{u}^k - \kappa(\nabla^T\xi^k + \mathbf{K}^T\boldsymbol{\zeta}^k) + \kappa[\beta\nabla^T(\mathbf{x}^k - \nabla\mathbf{u}^k) + \beta\mathbf{K}^T(\mathbf{b} + \mathbf{y}^k - \mathbf{Ku}^k)]$. Then, the solution $\mathbf{u}^k$ of (10) has the following closed form expression:

$$\mathbf{u}^{k+1} = \min(\mathbf{1}, \max(\mathbf{0}, \mathbf{g}^k)).$$

(ii) $\mathbf{v}$-subproblem. Subproblem (11) reduces to the following minimization problem:

$$\mathbf{v}^{k+1} = \arg\min_{\mathbf{0}\leq\mathbf{v}\leq\mathbf{1}} \ \frac{\beta}{2}\|\mathbf{v} \odot \mathbf{s}^k\|^2 - \langle\mathbf{v}, \mathbf{c}^k\rangle,$$

where $\mathbf{c}^k = \mathbf{1} - o \odot \boldsymbol{\pi}^k \odot |\mathbf{y}^k|$, $\mathbf{s}^k = \mathbf{o} \odot \mathbf{y}^k$. Therefore, the solution $\mathbf{v}^k$ can be computed as:

$$\mathbf{v}^{k+1} = \min(\mathbf{1}, \max(\mathbf{0}, \frac{\mathbf{c}^k}{\beta\mathbf{s}^k \odot \mathbf{s}^k})).$$

(iii) $(\mathbf{x}, \mathbf{y})$-subproblem. Variable $\mathbf{x}$ in Eq (12) is updated by solving the following problem:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^{2n}} \ \frac{\beta}{2}\|\mathbf{x} - \mathbf{h}^k\|^2 + \lambda\|\mathbf{x}\|_{p,1},$$

where $\mathbf{h}^k := -\nabla\mathbf{u}^{k+1} - \boldsymbol{\xi}^k/\beta$. It is not difficult to check that for $p = 1$,

$$\mathbf{x}^{k+1} = \text{sign}\big(\mathbf{h}^k\big) \odot \max\big(|\mathbf{h}^k| - \lambda/\beta, 0\big),$$

and when $p = 2$,

$$\begin{bmatrix} \mathbf{x}_i^{k+1} \\ \mathbf{x}_{i+n}^{k+1} \end{bmatrix} = \left(\max(0, 1 - \frac{\lambda/\beta}{\|(\mathbf{h}_i^k; \mathbf{h}_{i+n}^k)\|})\right) \begin{bmatrix} \mathbf{h}_i^k \\ \mathbf{h}_{i+n}^k \end{bmatrix}$$

Variable $\mathbf{y}$ in Eq (12) is updated by solving the following problem:

$$\mathbf{y}^{k+1} = \arg\min_{\mathbf{y}} \ \frac{\beta}{2}\|\mathbf{y} - \mathbf{q}^k\|^2 + \frac{\beta}{2}\|\mathbf{w}^k \odot |\mathbf{y}| + \boldsymbol{\pi}^k/\beta\|^2,$$

where $\mathbf{q}^k = \mathbf{Ku}^{k+1} - \mathbf{b} + \boldsymbol{\zeta}^k/\beta$ and $\mathbf{w}^k = \mathbf{o} \odot \mathbf{v}^{k+1}$. A simple computation yields that the solution $\mathbf{y}^k$ can be computed in closed form as:

$$\mathbf{y}^{k+1} = \text{sign}(\mathbf{q}^k) \odot \max\big(0, \frac{|\mathbf{q}^k| - \boldsymbol{\pi}^k \odot \mathbf{w}^k/\beta}{1 + \mathbf{v}^k \odot \mathbf{w}^k}\big),$$

The exposition above shows that the computation required in each iteration of Algorithm 1 is insignificant.

Proximal ADMM has excellent convergence in practice, but the optimization problem in Eq (8) is non-convex, so additional conditions are needed to guarantee convergence to a KKT point. Inspired by [38], we prove that under mild assumptions, our proximal ADMM algorithm always converges to a KKT point. Specifically, we have the following convergence result.

Theorem 1. ***Convergence of Algorithm 1.*** *Let* $X \triangleq (\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y})$ *and* $Y \triangleq (\boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{\pi})$. $\{X^k, Y^k\}_{k=1}^{\infty}$ *be the intermediate results of Algorithm 1 after the k-th iteration. Assume that* $\lim_{k\to\infty}(Y^{k+1} - Y^k) = 0$. *Then there exists a subsequence of* $\{X^k, Y^k\}$ *whose accumulation point satisfies the KKT conditions.*

Proof. Refer to the **supplementary material**. □

4. Experimental Validation

In this section, we provide empirical validation for our proposed $\ell_0 TV$-PADMM method by conducting extensive image denoising experiments and performing a thorough comparative analysis with the state-of-the-art. For more experimental results on image denoising and deblurring, please refer to the **supplementary material**.

In our experiments, we use 9 well-known test images of size 512 × 512. All code is implemented in MATLAB using a 3.20GHz CPU and 8GB RAM. Since past studies [7, 14] have shown that the isotropic TV model performs better than the anisotropic one, we choose $p = 2$ as the order of the TV norm here. In our experiments, we apply the following algorithms:

(i) $\ell_1 TV$**-SBM**, the Split Bregman Method (SBM) of [22], which has been implemented in [20]. We use this convex optimization method as our baseline implementation.

(ii) MFM, Median Filter Methods. We utilize adaptive median filtering to remove salt-and-pepper impulse noise and adaptive center-weighted median filtering to remove random-valued impulse noise.

(iii) TSM, the Two Stage Method[10, 11, 6]. The method first detects the damaged pixels by MFM and then solves the TV image inpainting problem.

(iv) $\ell_{02} TV$**-AOP**, the Adaptive Outlier Pursuit (AOP) method described in [42]. We use the implementation provided by the author. Here, we note that AOP iteratively calls the $\ell_1 TV$-*SBM* procedure, mentioned above.

(v) $\ell_0 TV$**-PDA**, the Penalty Decomposition Algorithm (PDA) [27] for solving the $\ell_0 TV$ optimization problem in Eq (6).

(vi) $\ell_0 TV$**-PADMM**, the proximal ADMM described in Algorithm 1 for solving the $\ell_0 TV$ optimization problem in Eq (6). Our MATLAB code is available online at `http://yuanganzhao.weebly.com/`.

4.1. Experiment Setup

For the image denoising task, we use the following strategy to generate noisy images. We corrupt the original image by injecting random-value and salt-and-pepper noise with different densities (10% to 70%). Then, we run all the previously mentioned algorithms on the generated noisy images. For $\ell_0 TV$-PADMM and $\ell_0 TV$-PDA, we use the same stopping criterion to terminate the optimization. For $\ell_1 TV$-SBM and $\ell_{02} TV$-AOP, we adopt the default stopping conditions provided by the authors. To evaluate these methods, we compute their Signal-to-Noise Ratios (SNRs). Since the corrupted pixels follow a Bernoulli-like distribution, it is generally hard to measure the data fidelity between the original images and the recovered images. Therefore, we consider three ways to measure SNR.

$$\begin{aligned} SNR_0(\boldsymbol{u}) &\triangleq \frac{n - \|\boldsymbol{u}^0 - \boldsymbol{u}^k\|_{0\text{-}\epsilon}}{n - \|\boldsymbol{u}^0 - \boldsymbol{u}^0\|_{0\text{-}\epsilon}}, \\ SNR_1(\boldsymbol{u}) &\triangleq 10\log_{10}\frac{\|\boldsymbol{u}^0 - \bar{\boldsymbol{u}}\|_1}{\|\boldsymbol{u}^k - \bar{\boldsymbol{u}}\|_1}, \\ SNR_2(\boldsymbol{u}) &\triangleq 10\log_{10}\frac{\|\boldsymbol{u}^0 - \bar{\boldsymbol{u}}\|_2^2}{\|\boldsymbol{u}^k - \bar{\boldsymbol{u}}\|_2^2}, \end{aligned}$$

where $\boldsymbol{u}^0$ is the original clean image and $\bar{\boldsymbol{u}}$ is the mean intensity value of $\boldsymbol{u}^0$, and $\|\cdot\|_{0\text{-}\epsilon}$ is the soft ℓ_0-norm which counts the number of elements whose magnitude is greater than a threshold ϵ. We adopt $\epsilon = \frac{20}{255}$ in our experiments.

4.2. Convergence of $\ell_0 TV$-PADMM

Here, we verify the convergence property of our $\ell_0 TV$-PADMM method by considering the 'cameraman' image subject to 30% random-valued impulse noise. We set $\lambda = 8$ for this problem. We record the objective and SNR values for $\ell_0 TV$-PADMM at every iteration k and plot these results in Figure 2.

We make two important observations from these results. **(i)**) The objective value (or the SNR value) does not necessarily decrease (or increase) monotonically, and we attribute this to the non-convexity

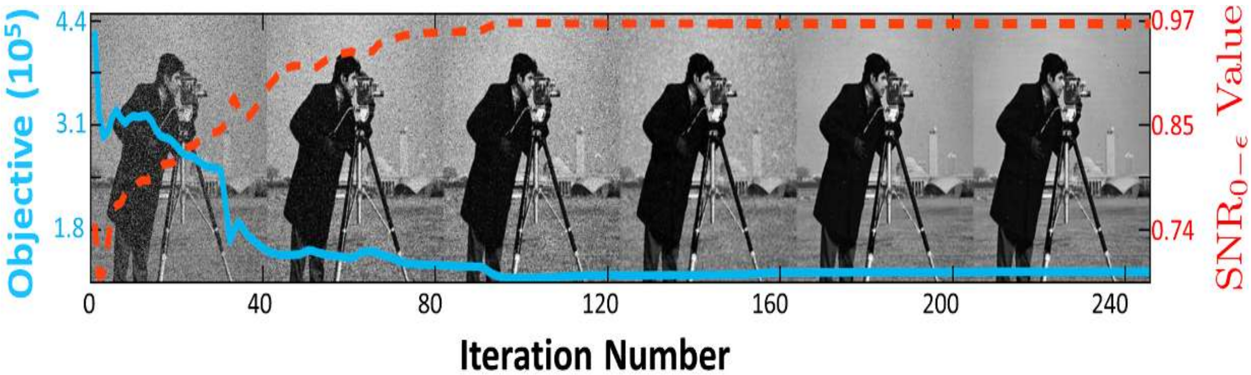

Figure 2: Asymptotic behavior for optimizing Eq (6) to denoise the corrupted 'cameraman' image. We plot the value of the objective function (solid blue line) and the SNR value (dashed red line) against the number of optimization iterations. At specific iterations (i.e. 1, 10, 20, 40, 80, and 160), we also show the denoised image. Clearly, the corrupting noise is being effectively removed throughout the optimization process.

of the optimization problem and the dynamic updates of the penalty factor in Algorithm 1. **(ii)** The objective and SNR values stabilize after the 200th iteration, which means that our algorithm has converged, and the increase of the SNR value is negligible after the 120th iteration. This implies that one may use a looser stopping criterion without sacrificing much restoration quality.

4.3. General Image Denoising Problems

In this subsection, we compare the performance of all 6 methods on general denoising problems. Table 3 shows image recovery results when random-value or salt-and-pepper impulse noise is added. We make the following interesting observations. **(i)** The $\ell_{02}TV$-AOP method greatly improves upon $\ell_1 TV$-SBM, MFM and TSM, by a large margin. These results are consistent with the reported results in [42]. **(ii)** The $\ell_0 TV$-PDA method outperforms $\ell_{02}TV$-AOP in most test cases because it adopts the ℓ_0-norm in the data fidelity term. **(iii)** In the case of random-value impulse noise, our $\ell_0 TV$-PADMM method is better than $\ell_0 TV$-PDA in SNR_0 value while it is comparable to $\ell_0 TV$-PDA in SNR_1 and SNR_2. On the other hand, when salt-and-pepper impulse noise is added, we find that $\ell_0 TV$-PADMM outperforms $\ell_0 TV$-PDA in most test cases. Interestingly, the performance gap between $\ell_0 TV$-PADMM and $\ell_0 TV$-PDA grows larger, as the noise level increases. **(iv)** For the same noise level, $\ell_0 TV$-PADMM achieves better recovery performance in the presence of salt-and-pepper impulse noise than random-valued impulse noise. This is primarily due to the fact that random-valued noise can take any value between 0 and 1, thus, making it more difficult to detect which pixels are corrupted.

5. Conclusions and Future Work

In this paper, we propose a new method for image restoration based on total variation (TV) with ℓ_0-norm data fidelity, which is particularly suitable for removing impulse noise. Although the resulting optimization model is non-convex, we design an efficient and effective proximal ADMM method for solving the equivalent MPEC problem of the original ℓ_0-norm minimization problem. Extensive numerical experiments indicate that the proposed ℓ_0TV model significantly outperforms the state-of-the-art in the presence of impulse noise. In particular, our proposed proximal ADMM solver is more effective than the penalty decomposition algorithm used for solving the ℓ_0TV problem [27] .

There are several research directions that are worthwhile to pursue for future work. One is to extend the present result to rank minimization problems. Another is to incorporate other priors into the ℓ_0-norm data fidelity for the problems of image/video recovery. The last is to apply the proposed MPEC-based proximal ADMM algorithm to other sparse optimization applications.

Acknowledgments

We would like to thank Prof. Shaohua Pan for her helpful discussions on the first draft of this paper. Yuan is supported by NSF-61402182. Research reported in this publication was supported by competitive research funding from King Abdullah University of Science and Technology (KAUST).

Table 3: General Denoising Problems. The results separated by '/' are SNR_0, SNR_1 and SNR_2, respectively.

Img. \ Alg.	ℓ_1TV-SBM	MFM	TSM	$\ell_{02}TV$-AOP	ℓ_0TV-PDA	ℓ_0TV-$PADMM$
Random-Value Impulse Noise						
walkbridge+10%	0.92/7.74/12.25	0.95/12.31/15.55	0.95/11.85/12.87	0.95/12.13/13.75	0.97/13.82/16.60	0.97/13.72/15.83
walkbridge+30%	0.82/6.17/10.37	0.89/8.62/11.01	0.85/5.84/7.82	0.89/7.78/11.47	0.91/9.66/12.81	0.91/9.27/11.67
walkbridge+50%	0.67/4.17/7.13	0.76/4.94/5.72	0.69/2.70/4.77	0.79/5.42/8.73	0.84/7.03/10.11	0.85/6.96/9.20
walkbridge+70%	0.46/2.07/3.55	0.56/1.95/1.74	0.50/1.26/2.16	0.59/3.02/4.97	0.65/4.00/6.20	0.76/5.12/7.04
walkbridge+90%	0.30/0.35/0.64	0.32/-0.21/-1.06	0.30/0.02/-0.00	0.30/0.41/0.75	0.34/0.75/1.30	0.56/2.63/3.89
pepper+10%	0.99/14.98/22.29	0.99/19.10/21.53	0.97/13.48/15.78	0.99/13.60/20.32	1.00/20.16/24.57	0.99/17.98/21.03
pepper+30%	0.97/11.57/16.61	0.96/12.25/13.61	0.87/6.28/9.52	0.98/12.04/16.81	0.98/15.09/19.70	0.98/14.50/18.14
pepper+50%	0.85/6.92/9.61	0.85/6.69/6.72	0.71/3.46/5.45	0.94/9.68/13.09	0.96/11.77/15.68	0.96/11.56/14.14
pepper+70%	0.59/3.02/4.25	0.63/2.76/2.10	0.52/1.63/2.38	0.79/5.19/6.21	0.84/6.77/8.87	0.93/8.93/11.34
pepper+90%	0.30/0.55/0.55	0.35/0.09/-1.03	0.31/0.29/0.09	0.35/0.89/1.00	0.39/1.33/1.72	0.76/4.25/4.81
mandrill+10%	0.92/7.40/6.95	0.89/8.08/8.96	0.93/9.64/9.64	0.93/9.64/9.64	0.95/10.83/11.01	0.95/10.38/9.65
mandrill+30%	0.76/3.81/5.88	0.83/5.95/6.64	0.83/4.74/4.92	0.85/5.83/6.78	0.86/6.65/7.23	0.86/6.42/6.46
mandrill+50%	0.65/2.90/4.59	0.73/3.63/3.70	0.69/1.99/3.36	0.74/3.62/5.02	0.77/4.63/5.54	0.78/4.39/4.47
mandrill+70%	0.51/1.58/2.51	0.57/1.35/0.65	0.52/0.95/1.49	0.62/2.33/3.44	0.64/2.87/3.93	0.70/3.08/3.44
mandrill+90%	0.37/0.24/0.45	0.36/-0.59/-1.90	0.34/-0.08/-0.35	0.39/0.54/0.91	0.42/0.78/1.22	0.58/1.91/2.49
lenna+10%	0.98/11.39/18.21	0.99/17.95/22.98	0.98/13.72/14.33	0.99/14.22/19.13	0.99/17.42/21.71	0.99/16.74/20.10
lenna+30%	0.96/9.69/15.25	0.96/11.50/13.87	0.89/6.27/9.84	0.98/10.73/16.55	0.98/13.08/17.68	0.97/12.62/15.99
lenna+50%	0.86/6.41/9.74	0.86/6.31/6.70	0.74/3.47/5.74	0.94/8.64/12.53	0.95/10.24/14.43	0.95/9.88/12.69
lenna+70%	0.61/2.86/4.23	0.65/2.51/1.97	0.54/1.64/2.56	0.78/4.99/6.97	0.85/6.42/9.19	0.91/7.61/9.83
lenna+90%	0.38/0.55/0.65	0.37/-0.17/-1.28	0.33/0.16/0.03	0.45/0.92/1.13	0.47/1.34/1.92	0.77/4.39/5.55
jetplane+10%	0.98/11.55/17.53	0.99/17.50/20.98	0.98/12.79/13.26	0.99/13.06/19.08	0.99/16.86/19.95	0.98/15.31/16.59
jetplane+30%	0.95/9.21/13.69	0.95/10.34/11.51	0.87/4.99/7.25	0.97/10.41/15.05	0.97/12.42/15.62	0.97/11.48/12.40
jetplane+50%	0.78/4.72/7.61	0.80/4.52/4.00	0.69/1.49/2.75	0.92/7.92/10.56	0.94/9.32/12.24	0.94/8.97/10.10
jetplane+70%	0.43/0.54/1.49	0.53/0.61/-0.69	0.47/-0.51/-0.51	0.67/3.24/4.76	0.74/4.36/6.37	0.89/6.63/7.29
jetplane+90%	0.31/-1.00/-1.25	0.25/-1.80/-3.65	0.26/-1.85/-2.85	0.14/-1.60/-2.18	0.26/-1.20/-1.49	0.73/3.30/3.53
cameraman+10%	0.98/13.62/20.50	0.99/20.12/24.95	0.98/14.19/15.65	0.99/14.73/21.81	0.99/18.80/23.21	0.98/17.25/19.45
cameraman+30%	0.93/10.00/14.60	0.95/12.29/14.17	0.87/6.67/9.81	0.97/12.35/17.52	0.97/14.22/17.82	0.97/12.86/15.13
cameraman+50%	0.74/5.56/7.71	0.82/6.44/6.78	0.69/3.50/5.43	0.91/9.09/12.04	0.94/10.79/14.08	0.95/10.90/12.56
cameraman+70%	0.59/2.70/3.38	0.59/2.67/2.32	0.49/1.76/2.44	0.68/3.68/4.19	0.74/5.23/6.81	0.90/8.52/10.01
cameraman+90%	0.34/0.78/0.72	0.34/0.29/-0.54	0.29/0.49/0.32	0.41/1.06/0.94	0.41/1.29/1.35	0.67/3.56/3.91
boat+10%	0.96/10.39/16.32	0.98/16.25/20.15	0.98/13.46/14.37	0.98/13.89/18.22	0.98/16.38/19.94	0.98/15.78/18.10
boat+30%	0.93/8.80/13.78	0.94/10.76/13.20	0.88/6.26/9.32	0.96/9.97/14.65	0.96/11.93/15.70	0.94/10.98/13.39
boat+50%	0.80/5.80/9.03	0.82/5.92/6.49	0.72/3.04/5.43	0.90/7.77/11.12	0.92/9.26/12.89	0.92/8.77/10.59
boat+70%	0.55/2.48/3.91	0.61/2.31/1.93	0.52/1.35/2.40	0.71/4.21/5.92	0.77/5.44/7.98	0.87/6.86/8.56
boat+90%	0.35/0.42/0.79	0.35/-0.12/-1.00	0.31/0.05/0.13	0.31/0.66/1.18	0.37/1.01/1.79	0.71/3.78/4.84
pirate+10%	0.93/10.06/15.58	0.97/14.97/18.50	0.96/13.26/14.26	0.97/13.26/17.13	0.97/15.66/18.60	0.97/15.46/17.78
pirate+30%	0.88/8.19/12.78	0.91/10.11/12.22	0.85/6.43/8.82	0.93/9.36/13.87	0.93/11.46/14.88	0.93/11.00/13.12
pirate+50%	0.65/4.69/7.27	0.76/5.53/6.00	0.67/3.16/4.92	0.83/6.95/10.28	0.87/8.64/11.83	0.89/8.70/10.60
pirate+70%	0.42/2.05/2.93	0.53/2.20/1.81	0.46/1.48/2.02	0.55/2.86/3.85	0.62/4.02/5.61	0.82/6.74/8.54
pirate+90%	0.26/0.36/0.12	0.29/0.05/-0.92	0.26/0.21/-0.14	0.28/0.46/0.25	0.31/0.74/0.66	0.51/2.26/2.40
house+10%	1.00/17.32/27.50	1.00/23.11/29.74	0.98/13.52/16.73	1.00/17.02/29.42	1.00/24.42/32.64	1.00/22.94/28.08
house+30%	0.98/13.65/20.27	0.97/13.79/15.43	0.89/6.74/10.24	0.99/14.71/23.67	1.00/18.78/26.78	0.99/17.70/23.57
house+50%	0.88/8.29/12.00	0.85/7.31/7.38	0.71/3.76/5.76	0.97/11.89/16.93	0.98/14.72/20.80	0.98/14.40/19.99
house+70%	0.49/2.99/4.25	0.61/3.01/2.35	0.50/1.83/2.51	0.78/6.06/8.20	0.85/7.82/10.83	0.95/11.20/14.70
house+90%	0.29/0.85/0.89	0.32/0.37/-0.78	0.29/0.48/0.21	0.38/1.04/1.01	0.36/1.41/1.84	0.83/6.17/6.82
Salt-and-Pepper Impulse Noise						
walkbridge+10%	0.91/7.54/12.36	0.96/12.88/17.26	0.98/15.83/19.88	0.98/15.84/19.88	0.99/17.16/22.66	0.99/17.48/23.16
walkbridge+30%	0.84/6.51/10.66	0.94/10.43/14.27	0.96/11.66/16.44	0.96/11.66/16.44	0.96/11.96/17.08	0.97/12.28/17.54
walkbridge+50%	0.76/5.04/7.80	0.89/8.12/11.42	0.92/9.31/13.96	0.92/9.30/13.94	0.92/9.15/13.84	0.93/9.52/14.30
walkbridge+70%	0.59/2.63/4.50	0.82/6.06/8.67	0.87/7.33/11.51	0.87/7.32/11.50	0.85/6.89/10.99	0.87/7.37/11.56
walkbridge+90%	0.36/1.10/1.94	0.67/3.66/5.08	0.73/4.78/7.76	0.73/4.77/7.73	0.57/3.29/5.81	0.74/4.82/7.82
pepper+10%	0.99/15.16/22.48	0.99/14.77/20.14	1.00/20.48/24.91	1.00/20.48/24.91	1.00/23.21/30.49	1.00/23.89/30.95
pepper+30%	0.97/11.91/16.29	0.98/14.60/18.35	0.99/16.84/22.92	0.99/16.85/22.94	0.99/17.69/24.78	1.00/18.46/25.59
pepper+50%	0.90/8.00/9.81	0.97/12.86/16.10	0.99/14.86/21.54	0.99/14.82/21.49	0.99/14.53/21.09	0.99/15.44/22.41
pepper+70%	0.69/4.06/6.20	0.95/10.56/13.34	0.98/12.48/18.52	0.98/12.47/18.49	0.97/11.39/16.90	0.98/12.69/18.73
pepper+90%	0.33/0.97/1.58	0.89/7.18/8.48	0.93/8.84/12.70	0.93/8.77/12.55	0.75/4.84/7.90	0.93/9.04/12.88
mandrill+10%	0.91/6.36/6.95	0.93/9.75/11.31	0.97/13.08/14.27	0.97/13.08/14.27	0.98/14.39/17.08	0.98/14.54/17.17
mandrill+30%	0.75/4.00/6.04	0.90/7.77/9.03	0.92/8.88/10.71	0.92/8.88/10.71	0.93/9.30/11.79	0.93/9.44/11.88
mandrill+50%	0.69/3.15/4.45	0.84/5.69/6.57	0.87/6.63/8.48	0.87/6.62/8.47	0.87/6.71/8.82	0.88/6.81/8.82
mandrill+70%	0.55/1.59/2.70	0.76/3.84/4.30	0.80/4.87/6.49	0.80/4.87/6.49	0.79/4.76/6.59	0.80/4.90/6.54
mandrill+90%	0.37/0.47/0.83	0.63/1.97/1.89	0.69/3.08/4.28	0.69/3.08/4.27	0.60/2.42/3.83	0.69/3.13/4.39
lenna+10%	0.98/11.44/18.12	1.00/16.18/23.62	1.00/20.52/28.42	1.00/20.52/28.42	1.00/20.74/29.04	1.00/21.07/29.89
lenna+30%	0.96/9.86/14.77	0.99/13.59/19.23	0.99/15.55/23.03	0.99/15.55/23.04	0.99/15.40/22.66	1.00/15.81/23.67
lenna+50%	0.90/7.11/9.39	0.97/11.16/15.78	0.99/12.92/19.87	0.99/12.92/19.87	0.98/12.49/18.93	0.99/13.07/20.10
lenna+70%	0.69/3.76/6.16	0.95/8.96/12.49	0.97/10.66/16.41	0.97/10.65/16.41	0.96/9.90/15.17	0.97/10.78/16.54
lenna+90%	0.42/1.02/1.61	0.87/6.01/7.47	0.91/7.55/11.09	0.91/7.53/11.03	0.75/4.63/7.70	0.92/7.70/11.27
jetplane+10%	0.98/11.64/17.60	1.00/16.96/23.40	1.00/20.38/26.82	1.00/20.38/26.82	1.00/20.73/28.03	1.00/21.31/29.20
jetplane+30%	0.95/9.60/13.67	0.98/13.56/17.86	0.99/15.47/21.87	0.99/15.47/21.88	0.99/15.27/21.63	0.99/15.90/22.74
jetplane+50%	0.89/6.96/9.03	0.96/10.89/14.14	0.98/12.72/18.45	0.98/12.72/18.45	0.98/12.07/17.27	0.98/12.87/18.53
jetplane+70%	0.71/3.22/4.02	0.93/8.48/10.50	0.96/10.16/14.57	0.96/10.16/14.57	0.94/9.18/13.27	0.96/10.28/14.57
jetplane+90%	0.42/0.70/1.02	0.87/5.61/6.01	0.89/6.57/8.64	0.89/6.56/8.60	0.54/2.44/4.85	0.89/6.80/8.70
cameraman+10%	0.98/13.76/20.61	1.00/19.83/27.21	1.00/22.43/30.74	1.00/22.43/30.74	1.00/22.71/29.79	1.00/23.83/32.14
cameraman+30%	0.94/10.60/14.03	0.99/15.91/21.06	1.00/17.93/25.93	0.99/17.90/25.84	0.99/17.43/24.24	1.00/18.56/26.55
cameraman+50%	0.84/6.71/9.22	0.97/13.08/17.21	0.99/15.07/21.87	0.99/15.05/21.83	0.98/14.16/19.99	0.99/15.31/22.04
cameraman+70%	0.65/4.00/5.31	0.94/10.56/13.48	0.97/12.14/17.25	0.97/12.14/17.23	0.95/11.06/15.47	0.97/12.31/17.26
cameraman+90%	0.48/1.50/1.49	0.88/7.51/8.45	0.90/8.35/11.20	0.90/8.34/11.19	0.70/4.68/7.47	0.90/8.67/11.34
boat+10%	0.96/10.40/16.28	0.99/15.31/21.41	0.99/19.02/24.93	0.99/19.02/24.93	1.00/20.01/27.23	1.00/20.39/28.04
boat+30%	0.93/8.90/13.31	0.97/12.76/17.46	0.99/14.53/20.95	0.99/14.54/20.96	0.99/14.65/21.10	0.99/15.09/22.00
boat+50%	0.86/6.57/8.82	0.95/10.40/14.23	0.97/12.02/17.86	0.97/12.01/17.86	0.97/11.67/17.23	0.98/12.23/18.12
boat+70%	0.66/3.52/5.45	0.91/8.19/10.97	0.95/9.69/14.38	0.95/9.69/14.38	0.93/9.04/13.55	0.95/9.79/14.42
boat+90%	0.51/1.52/1.98	0.82/5.54/6.80	0.86/6.69/9.34	0.86/6.66/9.27	0.68/4.04/7.16	0.87/6.85/9.41
pirate+10%	0.94/10.18/15.69	0.98/15.29/20.67	0.98/17.54/22.44	0.98/17.53/22.43	0.99/19.58/25.95	0.99/19.97/26.63
pirate+30%	0.90/8.66/12.90	0.96/12.58/16.77	0.97/13.80/19.47	0.97/13.76/19.39	0.98/14.23/19.98	0.98/14.66/20.69
pirate+50%	0.80/6.43/8.96	0.93/10.19/13.71	0.96/11.62/17.05	0.95/11.56/16.94	0.95/11.34/16.57	0.96/11.87/17.36
pirate+70%	0.58/3.21/5.49	0.87/7.99/10.56	0.92/9.48/14.10	0.92/9.46/14.07	0.89/8.78/13.22	0.92/9.56/14.20
pirate+90%	0.29/1.02/1.78	0.76/5.36/6.67	0.80/6.50/9.64	0.80/6.47/9.58	0.55/3.87/6.39	0.81/6.60/9.72
house+10%	1.00/17.45/27.43	1.00/23.13/33.35	1.00/26.60/40.15	1.00/26.58/40.11	1.00/26.96/40.08	1.00/28.08/42.12
house+30%	0.99/13.99/19.51	0.99/18.39/25.86	1.00/21.48/34.08	1.00/21.48/34.08	1.00/20.87/32.12	1.00/22.24/34.92
house+50%	0.92/9.32/12.44	0.99/15.25/21.58	1.00/18.21/29.65	1.00/18.21/29.65	1.00/17.14/26.82	1.00/18.76/30.02
house+70%	0.79/6.03/8.38	0.97/12.55/17.24	1.00/14.98/24.09	1.00/14.97/24.08	0.98/13.56/20.92	1.00/15.47/24.45
house+90%	0.33/1.74/2.66	0.92/9.13/10.95	0.95/10.43/15.48	0.95/10.35/15.24	0.73/5.60/9.26	0.96/11.06/15.95

References

[1] V. A. A. Tikhonov. *Solution of ill-posed problems*. Winston, Washington, DC, 1977.

[2] G. Aubert and J.-F. Aujol. A variational approach to removing multiplicative noise. *SIAM Journal on Applied Mathematics (SIAP)*, 68(4):925–946, 2008.

[3] J.-F. Aujol. Some first-order algorithms for total variation based image restoration. *Journal of Mathematical Imaging and Vision*, 34(3):307–327, 2009.

[4] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing (TIP)*, 18(11):2419–2434, 2009.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(11):1222–1239, 2001.

[6] J.-F. Cai, R. H. Chan, and M. Nikolova. Fast two-phase image deblurring under impulse noise. *Journal of Mathematical Imaging and Vision*, 36(1):46–53, Jan. 2010.

[7] J.-F. Cai, B. Dong, S. Osher, and Z. Shen. Image restoration: Total variation, wavelet frames, and beyond. *Journal of the American Mathematical Society (JAMS)*, 25(4):1033–1089, 2012.

[8] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97, 2004.

[9] A. Chambolle, D. Cremers, and T. Pock. A convex approach to minimal partitions. *SIAM Journal on Imaging Sciences (SIIMS)*, 5(4):1113–1158, 2012.

[10] R. H. Chan, C. Ho, and M. Nikolova. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Transactions on Image Processing (TIP)*, 14(10):1479–1485, 2005.

[11] R. H. Chan, C. Hu, and M. Nikolova. An iterative procedure for removing random-valued impulse noise. *IEEE SIGNAL PROCESSING LETTERS*, 11(12):921–924, 2004.

[12] T. F. Chan, G. H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM Journal on Scientific Computing (SISC)*, 20(6):1964–1977, 1999.

[13] R. Chartrand and V. Staneva. A quasi-newton method for total variation regularization of images corrupted by non-gaussian noise. *IET Image Processing*, 2:295–303, 2008.

[14] D.-Q. Chen, H. Zhang, and L.-Z. Cheng. A fast fixed point algorithm for total variation deblurring and segmentation. *Journal of Mathematical Imaging and Vision*, 43(3):167–179, 2012.

[15] C. Clason. ℓ_∞ fitting for inverse problems with uniform noise. *Inverse Problems*, 28(10):104007, 2012.

[16] C. Clason, B. Jin, and K. Kunisch. A duality-based splitting method for ℓ_1-tv image restoration with automatic regularization parameter choice. *SIAM Journal Scientific Computing (SISC)*, 32(3):1484–1505, 2010.

[17] A. d'Aspremont. A semidefinite representation for some minimum cardinality problems. In *IEEE Conference on Decision and Control (CDC)*, volume 5, pages 4985–4990, 2003.

[18] M. Feng, J. E. Mitchell, J.-S. Pang, X. Shen, and A. Wächter. Complementarity formulations of $\ell 0$-norm optimization problems. 2013.

[19] S. GEMAN and D. GEMAN. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 6(6):721–741, 1984.

[20] P. Getreuer. tvreg v2: Variational imaging methods for denoising, deconvolution, inpainting, and segmentation, `http://www.mathworks.com/matlabcentral/fileexchange/29743`. 2010.

[21] D. Goldfarb and W. Yin. Second-order cone programming methods for total variation-based image restoration. *SIAM Journal on Scientific Computing (SISC)*, 27(2):622–645, 2005.

[22] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences (SIIMS)*, 2(2):323–343, 2009.

[23] B. He, M. Xu, and X. Yuan. Solving large-scale least squares semidefinite programming by alternating direction methods. *SIAM Journal on Matrix Analysis and Applications (SIMAX)*, 32(1):136–152, 2011.

[24] J. Hu. On linear programs with linear complementarity constraints. pages 1–129, 2008.

[25] H. Ji, S. Huang, Z. Shen, and Y. Xu. Robust video restoration by joint sparse and low rank matrix approximation. *SIAM Journal on Imaging Sciences (SIIMS)*, 4(4):1122–1142, 2011.

[26] T. Le, R. Chartrand, and T. J. Asaki. A variational approach to reconstructing images corrupted by poisson noise. *Journal of Mathematical Imaging and Vision*, 27(3):257–263, 2007.

[27] Z. Lu and Y. Zhang. Sparse approximation via penalty decomposition methods. *SIAM Journal on Optimization (SIOPT)*, 23(4):2448–2478, 2013.

[28] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

[29] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, Apr. 1995.

[30] Y. E. Nesterov. *Introductory lectures on convex optimization: a basic course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2003.

[31] M. K. Ng, L. Qi, Y.-F. Yang, and Y.-M. Huang. On semismooth newton's methods for total variation minimization. *Journal of Mathematical Imaging and Vision*, 27(3):265–276, 2007.

[32] M. Nikolova and M. K. Ng. Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM Journal on Scientific Computing (SISC)*, 27(3):937–966, 2005.

[33] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[34] G. Steidl and T. Teuber. Removing multiplicative noise by douglas-rachford splitting methods. *Journal of Mathematical Imaging and Vision*, 36(2):168–184, 2010.

[35] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences (SIIMS)*, 1(3):248–272, 2008.

[36] P. Weiss, G. Aubert, and L. Blanc-Féraud. Some application of ℓ_∞ constraints in image processing. *INRIA Research Report*, 6115, 2006.

[37] P. Weiss, L. Blanc-Féraud, and G. Aubert. Efficient schemes for total variation minimization under constraints in image processing. *SIAM Journal Scientific Computing (SISC)*, 31(3):2047–2080, 2009.

[38] Z. Wen, C. Yang, X. Liu, and S. Marchesini. Alternating direction methods for classical and ptychographic phase retrieval. *Inverse Problems*, 28(11):115010, 2012.

[39] H. Woo and S. Yun. Proximal linearized alternating direction method for multiplicative denoising. *SIAM Journal on Scientific Computing (SISC)*, 35(2):B336–B358, 2013.

[40] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via ℓ_0 gradient minimization. *ACM Transactions on Graphics (TOG)*, 30(6):174, 2011.

[41] L. Xu, S. Zheng, and J. Jia. Unnatural ℓ_0 sparse representation for natural image deblurring. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

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

[43] J. Yang, Y. Zhang, and W. Yin. An efficient tvl1 algorithm for deblurring multichannel images corrupted by impulsive noise. *SIAM Journal on Scientific Computing (SISC)*, 31(4):2842–2865, 2009.

[44] X. Zhang, M. Burger, X. Bresson, and S. Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal on Imaging Sciences (SIIMS)*, 3(3):253–276, 2010.

[45] T. Zhou and D. Tao. Godec: Randomized low-rank & sparse matrix decomposition in noisy case. In *International Conference on Machine Learning (ICML)*, pages 33–40, 2011.

[46] W. Zuo and Z. Lin. A generalized accelerated proximal gradient approach for total-variation-based image restoration. *IEEE Transactions on Image Processing (TIP)*, 20(10):2748–2759, 2011.