

ℓ_0 TV: A New Method for Image Restoration in the Presence of Impulse Noise

Ganzhao Yuan¹ and Bernard Ghanem²

¹South China University of Technology (SCUT), P.R. China

²King 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_0 -norm data fidelity, only give sub-optimal performance. In this paper, we propose a new method, called ℓ_0 TV-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_0 TV-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_0 TV-PADMM to the problems of image denoising and deblurring in the presence of impulse noise. Our extensive experiments demonstrate that ℓ_0 TV-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:

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

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:

$$\min_{\mathbf{u}} \ell(\mathbf{K}\mathbf{u}, \mathbf{b}) + \lambda \Omega(\nabla_x \mathbf{u}, \nabla_y \mathbf{u}) \quad (2)$$

where $\ell(\mathbf{K}\mathbf{u}, \mathbf{b})$ measures the data fidelity between $\mathbf{K}\mathbf{u}$ and the observation \mathbf{b} and $\nabla_x \in \mathbb{R}^{n \times n}$ and $\nabla_y \in \mathbb{R}^{n \times n}$ are two suitable linear transformation matrices such that $\nabla_x \mathbf{u} \in \mathbb{R}^n$ and $\nabla_y \mathbf{u} \in \mathbb{R}^n$ compute the discrete gradients of the image u along the x -axis and y -axis, respectively¹, $\Omega(\nabla_x \mathbf{u}, \nabla_y \mathbf{u})$ is the regularizer on $\nabla_x \mathbf{u}$ and $\nabla_y \mathbf{u}$, and λ is a

¹In practice, one does not need to compute and store the matrices ∇_x and ∇_y explicitly. Since the adjoint of the gradient operator ∇ is the negative divergence operator $-\text{div}$, i.e., $\langle \mathbf{r}, \nabla_x \mathbf{u} \rangle = \langle -\text{div}_x \mathbf{r}, \mathbf{u} \rangle$, $\langle \mathbf{s}, \nabla_y \mathbf{u} \rangle = \langle -\text{div}_y \mathbf{s}, \mathbf{u} \rangle$ 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

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

Table 1: Regularization Models

Regularization Function	Desc. and Ref.
$\Omega_{\text{tik}}(\mathbf{g}, \mathbf{h}) = \sum_{i=1}^n \mathbf{g}_i^2 + \mathbf{h}_i^2$	Tikhonov-like, [1]
$\Omega_{\text{tv}_2}(\mathbf{g}, \mathbf{h}) = \sum_{i=1}^n (\mathbf{g}_i^2 + \mathbf{h}_i^2)^{\frac{1}{2}}$	isotropic, [33, 39]
$\Omega_{\text{tv}_1}(\mathbf{g}, \mathbf{h}) = \sum_{i=1}^n \mathbf{g}_i + \mathbf{h}_i $	anisotropic, [35, 43]
$\Omega_{\text{stv}}(\mathbf{g}, \mathbf{h}) = \sum_{i=1}^n (\mathbf{g}_i^2 + \mathbf{h}_i^2 + \epsilon^2)^{\frac{1}{2}}$	smooth TV, [12, 36]
$\Omega_{\text{hub}}(\mathbf{g}, \mathbf{h}) = \sum_{i=1}^n \varphi(\mathbf{g}_i; \mathbf{h}_i)$ $\varphi(\mathbf{g}_i; \mathbf{h}_i) = \begin{cases} \frac{\epsilon}{2} \ \mathbf{g}_i; \mathbf{h}_i\ _2^2; & \ \mathbf{g}_i; \mathbf{h}_i\ _2 \leq \frac{1}{\epsilon} \\ \ \mathbf{g}_i; \mathbf{h}_i\ _2 - \frac{1}{2\epsilon}; & \text{otherwise} \end{cases}$	Huber-like, [32]
$\Omega_{\text{pot}}(\mathbf{g}, \mathbf{h}) = \sum_{i=1}^n \mathbf{g}_i _0 + \mathbf{h}_i _0$	Potts model, [40, 41]

Table 2: Data Fidelity Models

Data Fidelity Function	Noise and Ref.
$\ell_2(\mathbf{Ku}, \mathbf{b}) = \ \mathbf{Ku} - \mathbf{b}\ _2^2$	add. Gaussian [33, 8]
$\ell_1(\mathbf{Ku}, \mathbf{b}) = \ \mathbf{Ku} - \mathbf{b}\ _1$	add. Laplace [43, 16]
$\ell_\infty(\mathbf{Ku}, \mathbf{b}) = \ \mathbf{Ku} - \mathbf{b}\ _\infty$	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]
$\ell_{02}(\mathbf{Ku}, \mathbf{b}) = \ \mathbf{Ku} - \mathbf{b} + \mathbf{z}\ _2^2$ s.t. $\ \mathbf{z}\ _0 \leq k$	mixed Gaussian impulse [42, 45]
$\ell_0(\mathbf{Ku}, \mathbf{b}) = \ \mathbf{Ku} - \mathbf{b}\ _0$	add./mult. impulse [13], [this paper]

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.

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].

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-hard² 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_0 TV 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.

²The 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 non-zero 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: $p(\mathbf{u}) = \frac{1}{\vartheta} \exp(-\sigma \cdot \Omega_{tv}(\nabla_x \mathbf{u}, \nabla_y \mathbf{u}))$, where ϑ is a normalization factor, $\sigma \Omega_{tv}(\nabla_x \mathbf{u}, \nabla_y \mathbf{u})$ 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 ℓ_0 TV model:

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

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 $(\nabla_x \mathbf{u})_i$ and $(\nabla_y \mathbf{u})_i$ denote the i th component of the vectors $\nabla_x \mathbf{u}$ and $\nabla_y \mathbf{u}$, respectively. For convenience, we define $\forall \mathbf{x} \in \mathbb{R}^{2n}$:

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

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

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

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 ℓ_0 TV-PADMM method for solving Eq (6) in Figure 1.

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*

$$\|\mathbf{w}\|_0 = \min_{\mathbf{o} \leq \mathbf{v} \leq \mathbf{1}} \langle \mathbf{1}, \mathbf{1} - \mathbf{v} \rangle, \text{ s.t. } \mathbf{v} \odot |\mathbf{w}| = \mathbf{0}, \quad (7)$$

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**. \square

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

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

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

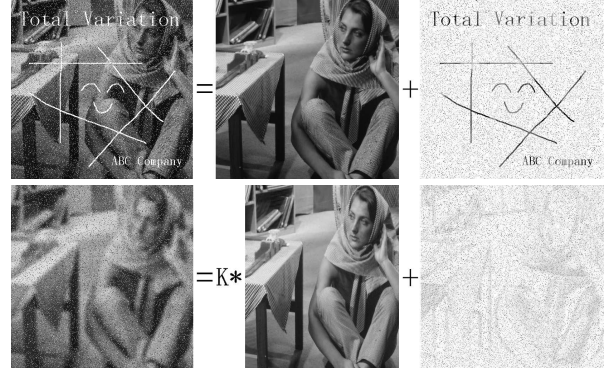


Figure 1: An example of an image recovery result using our proposed ℓ_0 TV-PADMM method. Left column: corrupted image. Middle column: recovered image. Right column: absolute residual between these two images.

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:

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

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

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

where ξ , ζ and π are the Lagrange multipliers associated with the constraints $\nabla \mathbf{u} = \mathbf{x}$, $\mathbf{K}\mathbf{u} - \mathbf{b} = \mathbf{y}$ 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 $(\mathbf{u}^0, \mathbf{v}^0, \mathbf{x}^0, \mathbf{y}^0, \xi^0, \zeta^0)$. Set $k = 0$. Select the parameters $\beta = 1$ and $\kappa \in (0, \frac{1}{\beta\|\nabla\|^2 + \beta\|\mathbf{K}\|^2})$.

(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:

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

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

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

(S.3) if $(k \text{ is a multiple of } 30)$, 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) **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

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

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 $\mathbf{g}^k = \mathbf{u}^k - \kappa(\nabla^T \xi^k + \mathbf{K}^T \zeta^k) + \kappa[\beta \nabla^T(\mathbf{x}^k - \nabla \mathbf{u}^k) + \beta \mathbf{K}^T(\mathbf{b} + \mathbf{y}^k - \mathbf{K}\mathbf{u}^k)]$. 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) **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 $\mathbf{c}^k = \mathbf{1} - \mathbf{o} \odot \pi^k \odot |\mathbf{y}^k|$, $\mathbf{s}^k = \mathbf{o} \odot \mathbf{y}^k$. 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) **(x, 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 $\mathbf{h}^k := -\nabla \mathbf{u}^{k+1} - \xi^k/\beta$. It is not difficult to check that for $p = 1$,

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

and when $p = 2$,

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

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

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

where $\mathbf{q}^k = \mathbf{K}\mathbf{u}^{k+1} - \mathbf{b} + \zeta^k/\beta$ 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:

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

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 $Y \triangleq (\xi, \zeta, \pi)$. $\{X^k, Y^k\}_{k=1}^\infty$ be the intermediate results of Algorithm 1 after the k -th iteration. Assume that $\lim_{k \rightarrow \infty} (Y^{k+1} - Y^k) = 0$. Then there exists a subsequence of $\{X^k, Y^k\}$ whose accumulation point satisfies the KKT conditions.*

Proof. Refer to the **supplementary material**. \square

4. Experimental Validation

In this section, we provide empirical validation for our proposed ℓ_0TV -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) ℓ_1TV -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) MFMM, 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 MFMM 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 ℓ_1TV -SBM procedure, mentioned above.

(v) ℓ_0TV -PDA, the Penalty Decomposition Algorithm (PDA) [27] for solving the ℓ_0TV optimization problem in Eq (6).

(vi) ℓ_0TV -PADMM, the proximal ADMM described in Algorithm 1 for solving the ℓ_0TV 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 ℓ_0TV -PADMM and ℓ_0TV -PDA, we use the same stopping criterion to terminate the optimization. For ℓ_1TV -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.

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

where \mathbf{u}^0 is the original clean image and $\bar{\mathbf{u}}$ is the mean intensity value of \mathbf{u}^0 , and $\|\cdot\|_{0-\epsilon}$ 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 ℓ_0TV -PADMM

Here, we verify the convergence property of our ℓ_0TV -PADMM method by considering the ‘camera-man’ image subject to 30% random-valued impulse noise. We set $\lambda = 8$ for this problem. We record the objective and SNR values for ℓ_0TV -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

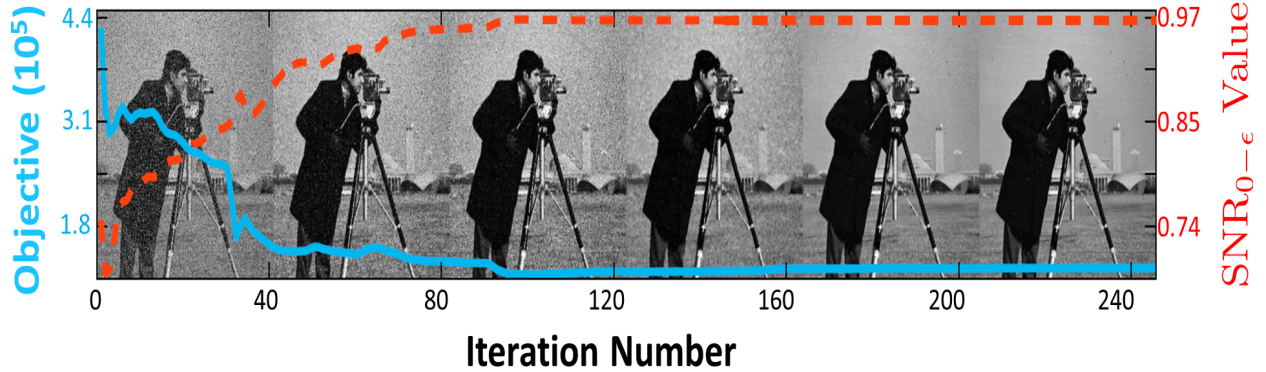


Figure 2: Asymptotic behavior for optimizing Eq (6) to denoise the corrupted ‘cameraman’ image. We plot the value of the objective function (solid blue line) and the SNR value (dashed red line) against the number of optimization iterations. At specific iterations (i.e. 1, 10, 20, 40, 80, and 160), we 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 ℓ_0TV -AOP method greatly improves upon ℓ_1TV -SBM, MFM and TSM, by a large margin. These results are consistent with the reported results in [42]. (ii) The ℓ_0TV -PDA method outperforms ℓ_0TV -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 ℓ_0TV -PADMM method is better than ℓ_0TV -PDA in SNR_0 value while it is comparable to ℓ_0TV -PDA in SNR_1 and SNR_2 . On the other hand, when salt-and-pepper impulse noise is added, we find that ℓ_0TV -PADMM outperforms ℓ_0TV -PDA in most test cases. Interestingly, the performance gap between ℓ_0TV -PADMM and ℓ_0TV -PDA grows larger, as the noise level increases. (iv) For the same noise level, ℓ_0TV -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.

Alg.	Img.	$\ell_1 TV-SBM$	MFM	TSM	$\ell_0 TV-AOP$	$\ell_0 TV-PDA$	$\ell_0 TV-PADMM$
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%	0.43/1.54/1.49	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%	0.31/0.01/-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%	0.34/0.78/0.70	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%	1.00/23.12/27.50	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	0.85/6.90/10.99	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	1.00/21.37/29.89	
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		

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 ℓ_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 ℓ_1 -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.

ℓ_0 TV: A New Method for Image Restoration in the Presence of Impulse Noise (Supplementary Material)

Ganzhao Yuan¹ and Bernard Ghanem²

¹South China University of Technology (SCUT), P.R. China

²King Abdullah University of Science and Technology (KAUST), Saudi Arabia

yuanganzhao@gmail.com, bernard.ghanem@kaust.edu.sa

The supplementary material is organized as follows. Section 1 presents the details of our proofs. Section 2 discusses the connection with existing work. Section 3 presents some additional experimental results.

1. Proofs

1.1. Proof of Lemma 1

Here, we prove the variational formulation of the ℓ_0 -norm.

Proof. This lemma is very natural. The total number of zero elements in w can be computed as

$$n - \|\mathbf{w}\|_0 = \max_{\mathbf{v} \in \{0,1\}} \sum_{i=1}^n \mathbf{v}_i, \text{ s.t. } \mathbf{v} \in \Phi \quad (1)$$

where $\Phi \triangleq \{\mathbf{v} \mid \mathbf{v}_i \odot |\mathbf{w}_i| = 0, \forall i \in [n]\}$. Note that when $\mathbf{w}_i = 0$, $\mathbf{v}_i = 1$ will be achieved by maximization, when $\mathbf{w}_i \neq 0$, $\mathbf{v}_i = 0$ will be enforced by the constraint. Thus, $\mathbf{v}_i = 1 - \text{sign}(|\mathbf{w}_i|)$. Since the objective function in Eq (1) is linear, maximization is always achieved at the boundaries of the feasible solution space. Thus, the constraint of $\mathbf{v}_i \in \{0,1\}$ can be relaxed to $0 \leq \mathbf{v}_i \leq 1$, we have:

$$\begin{aligned} \|\mathbf{w}\|_0 &= n - \max_{0 \leq \mathbf{v} \leq 1} \sum_{i=1}^n \mathbf{v}_i, \text{ s.t. } \mathbf{v} \in \Phi \\ &= \min_{0 \leq \mathbf{v} \leq 1} \langle \mathbf{1}, \mathbf{1} - \mathbf{v} \rangle, \text{ s.t. } \mathbf{v} \in \Phi \end{aligned}$$

□

1.2. Proof of Convergence of Algorithm 1

The global convergence of ADMM for convex problems was given by He and Yuan in [5] under the variation inequality framework. However, since our

optimization problem is non-convex, the convergence analysis for ADMM needs additional conditions. In non-convex optimization, convergence to a stationary point (local minimum) is the best convergence property that we can hope for. By imposing some conditions, Wen *et al.* [8] managed to show that the sequence generated by ADMM converges to a KKT point. In this section, along a similar line, we establish the convergence property of proximal ADMM.

First of all, we present the first-order KKT conditions of our ℓ_0 TV optimization problem. For simplicity, we define:

$$\Delta \triangleq \{z \mid \mathbf{0} \leq \mathbf{z} \leq \mathbf{1}\}. \quad (2)$$

Based on the augmented Lagrangian function of the ℓ_0 TV optimization problem, we naturally derive the following KKT conditions of the optimization problem for $\{\mathbf{u}^*, \mathbf{v}^*, \mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\xi}^*, \boldsymbol{\zeta}^*, \boldsymbol{\pi}^*\}$:

$$\begin{aligned} 0 &\leq \langle \nabla^T \boldsymbol{\xi}^* + \mathbf{K}^T \boldsymbol{\zeta}^*, \mathbf{u} - \mathbf{u}^* \rangle, \forall \mathbf{u} \in \Delta \\ 0 &\leq \langle \boldsymbol{\pi}^* \odot \mathbf{o} \odot |\mathbf{y}^*| - \mathbf{1}, \mathbf{v} - \mathbf{v}^* \rangle, \forall \mathbf{v} \in \Delta \\ 0 &\in \partial \lambda \|\mathbf{x}^*\|_{p,1} - \boldsymbol{\xi}^* \\ 0 &\in \boldsymbol{\pi}^* \odot \mathbf{v}^* \odot \mathbf{o} \odot \partial \|\mathbf{y}^*\|_1 - \boldsymbol{\zeta}^* \\ 0 &= \nabla \mathbf{u}^* - \mathbf{x}^* \\ 0 &= \mathbf{K} \mathbf{u}^* - \mathbf{b} - \mathbf{y}^* \\ 0 &= \mathbf{o} \odot \mathbf{v}^* \odot |\mathbf{y}^*| \end{aligned} \quad (3)$$

whose existence can be guaranteed by Robinson's constraint qualification. The following theorem establishes the convergence properties of the proposed algorithm, under the assumption that the iterates generated by Algorithm 1 exhibit no jumping behavior. Note that a similar condition was used in [8].

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

Proof. (i) $\{\xi, \zeta, \pi\}$ -subproblem. By the limit of $\{\xi^k, \zeta^k, \pi^k\}$ in the assumption and the update rule of $\{\xi^{k+1}, \zeta^{k+1}, \pi^{k+1}\}$ in Algorithm 1, we obtain

$$\lim_{k \rightarrow \infty} \nabla \mathbf{u}^{k+1} - \mathbf{x}^{k+1} = 0, \quad (4)$$

$$\lim_{k \rightarrow \infty} \mathbf{K} \mathbf{u}^{k+1} - \mathbf{b} - \mathbf{y}^{k+1} = 0, \quad (5)$$

$$\lim_{k \rightarrow \infty} \mathbf{o} \odot \mathbf{v}^{k+1} \odot |\mathbf{y}^{k+1}| = 0 \quad (6)$$

(ii) \mathbf{y} -subproblem. By the limit of ζ^k and π^k , and the update rule of \mathbf{y}^{k+1} in Algorithm 1, we have:

$$\begin{aligned} \lim_{k \rightarrow \infty} \{\mathbf{y}^{k+1} \in \arg \min_{\mathbf{y}} \frac{\beta}{2} \|\mathbf{K} \mathbf{u}^{k+1} - \mathbf{b} + \zeta^{k+1} / \beta - \mathbf{y}\|^2 \\ + \langle |\mathbf{y}|, \mathbf{o} \odot \mathbf{v}^{k+1} \odot \pi^{k+1} \rangle + \frac{\beta}{2} \|\mathbf{o} \odot \mathbf{v}^{k+1} \odot \mathbf{y}\|^2\} \end{aligned}$$

which is equivalent to:

$$\lim_{k \rightarrow \infty} -\zeta^{k+1} + \pi^{k+1} \odot \mathbf{v}^{k+1} \odot \mathbf{o} \odot \partial \|\mathbf{y}^{k+1}\|_1 = 0 \quad (7)$$

Moreover, we have the following limit:

$$\lim_{k \rightarrow \infty} \mathbf{y}^{k+1} - \mathbf{y}^k = 0. \quad (8)$$

(iii) \mathbf{x} -subproblem. By the limit of ξ^k and the update rule of \mathbf{x}^{k+1} , we have:

$$\begin{aligned} \lim_{k \rightarrow \infty} \mathbf{x}^{k+1} \in \arg \min_{\mathbf{x} \in \mathbb{R}^{2n}} \lambda \|\mathbf{x}\|_{p,1} \\ + \frac{\beta}{2} \|\mathbf{x} - (-\nabla \mathbf{u}^{k+1} - \beta^{-1} \xi^{k+1})\|^2. \end{aligned}$$

which is equivalent to:

$$\lim_{k \rightarrow \infty} -\xi^{k+1} + \partial \lambda \|\mathbf{x}\|_{p,1} = 0 \quad (9)$$

Clearly, we obtain the following limit:

$$\lim_{k \rightarrow \infty} \mathbf{x}^{k+1} - \mathbf{x}^k = 0. \quad (10)$$

(iv) \mathbf{v} -subproblem. By the limit of \mathbf{y}^k and π^k , we have:

$$\begin{aligned} \lim_{k \rightarrow \infty} \mathbf{v}^{k+1} \in \arg \min_{0 \leq \mathbf{v} \leq \mathbf{1}} \frac{\beta}{2} \|\mathbf{v} \odot \mathbf{o} \odot \mathbf{y}^{k+1}\|^2 \\ - \langle \mathbf{v}, \mathbf{1} - \mathbf{o} \odot \pi^{k+1} \odot |\mathbf{y}^{k+1}| \rangle, \end{aligned}$$

which is equivalent to:

$$\lim_{k \rightarrow \infty} \langle \pi^{k+1} \odot \mathbf{o} \odot |\mathbf{y}^{k+1}| - \mathbf{1}, \mathbf{v} - \mathbf{v}^{k+1} \rangle \geq 0, \quad \forall \mathbf{v} \in \Delta \quad (11)$$

(v) \mathbf{u} -subproblem. By the optimality of \mathbf{u}^{k+1} for the \mathbf{u} -subproblem, we have:

$$\begin{aligned} \forall \mathbf{u} \in \Delta, \quad 0 \leq \langle \nabla^T \xi^k + \beta \nabla^T (\nabla \mathbf{u}^k - \mathbf{x}^k) + \mathbf{K}^T \zeta^k + \\ \beta \mathbf{K}^T (\mathbf{K} \mathbf{u}^k - \mathbf{b} - \mathbf{y}^k) + \mathbf{D}(\mathbf{u}^{k+1} - \mathbf{u}^k), \mathbf{u} - \mathbf{u}^{k+1} \rangle \end{aligned}$$

Take the limit of the equality constraints, we have:

$$\begin{aligned} \lim_{k \rightarrow \infty} \langle \nabla^T \xi^k + \mathbf{K}^T \zeta^k + \mathbf{D}(\mathbf{u}^{k+1} - \mathbf{u}^k), \mathbf{u} - \mathbf{u}^{k+1} \rangle \geq 0, \\ \forall \mathbf{u} \in \Delta. \quad (12) \end{aligned}$$

On the other hand, it is easy to validate that the function $\mathcal{L}_0(\mathbf{u}, \mathbf{v}^k, \mathbf{x}^k, \mathbf{y}^k, \xi, \zeta, \pi^k)$ is jointly convex with respect to $\{\mathbf{u}, \xi, \zeta\}$. We define:

$$\mathbf{w} \triangleq \begin{pmatrix} \mathbf{u} \\ \xi \\ \zeta \end{pmatrix} \quad \text{and} \quad F(\mathbf{w}) \triangleq \begin{pmatrix} \nabla^T \xi - \mathbf{K}^T \zeta \\ \nabla \mathbf{u} - \mathbf{x}^k \\ \mathbf{K} \mathbf{u} - \mathbf{b} - \mathbf{y}^k \end{pmatrix}.$$

Notice that the mapping F is monotone by convexity. It follows that

$$\langle \mathbf{w}^{k+1} - \mathbf{w}^*, F(\mathbf{w}^{k+1}) \rangle \geq \langle \mathbf{w}^{k+1} - \mathbf{w}^*, F(\mathbf{w}^*) \rangle \geq 0, \quad (13)$$

From Eq (8), Eq(10) and the first inequality in Eq(13), we have:

$$\begin{aligned} \lim_{k \rightarrow \infty} \langle \mathbf{u}^{k+1} - \mathbf{u}^*, \nabla^T (\xi^{k+1} - \xi^*) + \mathbf{K}^T (\zeta^{k+1} - \zeta^*) \rangle + \\ \langle \xi^{k+1} - \xi^*, \nabla (\mathbf{u}^{k+1} - \mathbf{u}^*) \rangle + \langle \zeta^{k+1} - \zeta^*, \mathbf{K}(\mathbf{u}^{k+1} - \mathbf{u}^*) \rangle \geq 0 \end{aligned}$$

which can be simplified as:

$$\lim_{k \rightarrow \infty} 2 \langle \mathbf{u}^{k+1} - \mathbf{u}^*, \nabla^T (\xi^{k+1} - \xi^*) + \mathbf{K}^T (\zeta^{k+1} - \zeta^*) \rangle \geq 0 \quad (14)$$

By Eq (13), it holds that $\langle \mathbf{w}^{k+1} - \mathbf{w}^*, F(\mathbf{w}^{k+1}) \rangle \geq 0$, then we have:

$$\lim_{k \rightarrow \infty} \langle \mathbf{u}^{k+1} - \mathbf{u}^*, \nabla^T \xi^* - \mathbf{K}^T \zeta^* \rangle \geq 0 \quad (15)$$

Combining Eq (12), Eq (14) and Eq (15), we have:

$$\lim_{k \rightarrow \infty} \langle \mathbf{u}^{k+1} - \mathbf{u}^*, \mathbf{D}(\mathbf{u}^k - \mathbf{u}^{k+1}) \rangle \geq 0 \quad (16)$$

From the Pythagoras relation¹ and the inequality above, as $k \rightarrow \infty$ it follows that

$$\begin{aligned} \|\mathbf{u}^k - \mathbf{u}^*\|_{\mathbf{D}}^2 &= \|\mathbf{u}^{k+1} - \mathbf{u}^*\|_{\mathbf{D}}^2 + \|\mathbf{u}^k - \mathbf{u}^{k+1}\|_{\mathbf{D}}^2 + \\ &\quad 2 \langle \mathbf{u}^{k+1} - \mathbf{u}^*, \mathbf{D}(\mathbf{u}^k - \mathbf{u}^{k+1}) \rangle \\ &\geq \|\mathbf{u}^{k+1} - \mathbf{u}^*\|_{\mathbf{D}}^2 + \|\mathbf{u}^k - \mathbf{u}^{k+1}\|_{\mathbf{D}}^2 \quad (17) \\ &\geq \|\mathbf{u}^{k+1} - \mathbf{u}^*\|_{\mathbf{D}}^2 + 0. \quad (18) \end{aligned}$$

¹Pythagoras relation: $\|\mathbf{b} - \mathbf{a}\|^2 = \|\mathbf{c} - \mathbf{a}\|^2 + \|\mathbf{b} - \mathbf{c}\|^2 + 2 \langle \mathbf{c} - \mathbf{a}, \mathbf{b} - \mathbf{c} \rangle$

Together with the strict positive definiteness of \mathbf{D} , Eq (18) implies that the sequences $\{\|\mathbf{u}^k - \mathbf{u}^*\|_{\mathbf{D}}\}$ is monotone non-increasing. Moreover, the sequence $\{\|\mathbf{u}^k - \mathbf{u}^*\|_{\mathbf{D}}\}$ and $\{\mathbf{u}^k\}$ are bounded. On the other hand, from Eq (17), we have:

$$\begin{aligned}\|\mathbf{u}^k - \mathbf{u}^{k+1}\|_{\mathbf{D}}^2 &\leq \|\mathbf{u}^k - \mathbf{u}^*\|_{\mathbf{D}}^2 - \|\mathbf{u}^{k+1} - \mathbf{u}^*\|_{\mathbf{D}}^2 \\ &\leq \|\mathbf{u}^k - \mathbf{u}^*\|_{\mathbf{D}}^2 + 0\end{aligned}$$

which implies that the sequences $\|\mathbf{u}^k - \mathbf{u}^{k+1}\|_{\mathbf{D}}^2$ is also monotone non-increasing.

We denote $C = \|\mathbf{u}^0 - \mathbf{u}^*\|_{\mathbf{D}}^2 - \|\mathbf{u}^k - \mathbf{u}^*\|_{\mathbf{D}}^2$. Summing Eq (17) over $i = 0, 1, \dots, k$, we have:

$$\begin{aligned}C &\geq \sum_{i=0}^k \|\mathbf{u}^i - \mathbf{u}^{i+1}\|_{\mathbf{D}}^2 \\ &\geq (k+1)\|\mathbf{u}^k - \mathbf{u}^{k+1}\|_{\mathbf{D}}^2\end{aligned}$$

Therefore, we have $\lim_{k \rightarrow \infty} \|\mathbf{u}^k - \mathbf{u}^{k+1}\|_{\mathbf{D}}^2 = \frac{C}{k+1} = 0$. By the strict positive definiteness of \mathbf{D} , we have $\lim_{k \rightarrow \infty} \|\mathbf{u}^k - \mathbf{u}^{k+1}\| = 0$.

Notice that Eq (12) holds for each $\mathbf{u} \in \Delta$. Taking the limit $k \rightarrow \infty$ with $\mathbf{u} \in \Delta$ to this inequality, we obtain that

$$\begin{aligned}\forall \mathbf{u} \in \Delta, \quad &\langle \nabla^T \xi^k + \mathbf{K}^T \zeta^k, \mathbf{u} - \mathbf{u}^{k+1} \rangle \\ &\geq \langle \mathbf{D}(\mathbf{u}^{k+1} - \mathbf{u}^k), \mathbf{u}^{k+1} - \mathbf{u} \rangle \\ &\geq -\|\mathbf{D}(\mathbf{u}^{k+1} - \mathbf{u}^k)\| \|\mathbf{u}^{k+1} - \mathbf{u}\| \\ &= 0\end{aligned}\tag{19}$$

where the last inequality holds by the Cauchy-Schwarz Inequality. Based on Eqs (4,5,6,7,9,11,19), we conclude that as $k \rightarrow \infty$, there exists a subsequence of $\{X^k, Y^k\}$ whose accumulation point satisfies the KKT conditions in Eq (3). \square

2. Discussions on the connection with Existing Work

In this section, we discuss the connection between the proposed method ℓ_0TV -PADMM and prior work.

2.1. Connection with convex optimization method ℓ_1TV

The goal of image restoration in the presence of impulse noise has been pursued by a number of authors (see, e.g., [13, 3]) using ℓ_1TV , which can be formulated as follows:

$$\min_{\mathbf{0} \leq \mathbf{u} \leq 1} \|\mathbf{K}\mathbf{u} - \mathbf{b}\|_1 + \lambda \|\nabla \mathbf{u}\|_{p,1}, \tag{20}$$

It is generally believed that ℓ_1TV is able to remove the impulse noise properly. This is because ℓ_1 -norm provides the tightest convex relaxation for the ℓ_0 -norm over the unit ball in the sense of ℓ_∞ -norm. It is shown in [2] that the problem of minimizing $\|\mathbf{K}\mathbf{u} - \mathbf{b}\|_1$ is equivalent to $\|\mathbf{K}\mathbf{u} - \mathbf{b}\|_0$ with high probability under the assumptions that (i) $\mathbf{K}\mathbf{u} - \mathbf{b}$ is sparse at the optimal solution \mathbf{u}^* and (ii) \mathbf{K} is a random Gaussian matrix and sufficiently “incoherent” (i.e., number of rows in \mathbf{K} is greater than its number of columns). However, these two assumptions required in [2] do not necessarily hold true for our ℓ_0TV optimization problem. Specifically, when the noise level of the impulse noise is high, $\mathbf{K}\mathbf{u} - \mathbf{b}$ may not be sparse at the optimal solution \mathbf{u}^* . Moreover, the matrix \mathbf{K} is a square identity or ill-conditioned matrix. Generally, ℓ_1TV will only lead to a sub-optimal solution.

2.2. Connection with sparse plus low-rank matrix decomposition

Sparse plus low-rank matrix decomposition [9, 15, 6] is becoming a powerful tool that effectively corrects large errors in structured data in the last decade. It aims at decomposing a given corrupted image \mathbf{B} (which is of matrix form) into its sparse component (\mathbf{S}) and low-rank component (\mathbf{L}) by solving

$$\min_{\mathbf{B}, \mathbf{L}} \|\mathbf{S}\|_0 + \lambda \text{rank}(\mathbf{L}), \text{ s.t. } \mathbf{B} = \mathbf{L} + \mathbf{S}.$$

Here the sparse component represents the foreground of an image which can be treated as outliers or impulse noise, while the low-rank component corresponds to the background, which is highly correlated. This is equivalent to the following optimization problem:

$$\min_{\mathbf{L}} \|\mathbf{B} - \mathbf{L}\|_0 + \lambda \text{rank}(\mathbf{L}),$$

which is also based on ℓ_0 -norm data fidelity. While they consider the low-rank prior in their objective function, we consider the Total Variation (TV) prior in ours.

2.3. Connection with the Adaptive Outlier Pursuit algorithm

Very recently, Yan [12] proposed the following new model for image restoration in the presence of impulse noise and mixed Gaussian impulse noise:

$$\min_{\mathbf{u}, \mathbf{z}} \|\mathbf{K}\mathbf{u} - \mathbf{b} - \mathbf{z}\|_2^2 + \lambda \|\nabla \mathbf{u}\|_{p,1}, \text{ s.t. } \|\mathbf{z}\|_0 \leq k. \tag{21}$$

They further reformulate the problem above into

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{v}} \quad & \|\mathbf{v} \odot (\mathbf{K}\mathbf{u} - \mathbf{b})\|_2^2 + \lambda \|\nabla \mathbf{u}\|_{p,1}, \\ \text{s.t.} \quad & \mathbf{0} \leq \mathbf{v} \leq \mathbf{1}, \quad \langle \mathbf{v}, \mathbf{1} \rangle \leq n - k \end{aligned}$$

and then solve this problem using an Adaptive Outlier Pursuit(AOP) algorithm. The AOP algorithm is actually an alternating minimization method, which separates the minimization problem over u and v into two steps. By iteratively restoring the images and updating the set of damaged pixels, it is shown that AOP algorithm outperforms existing state-of-the-art methods for impulse noise denoising, by a large margin.

Despite the merits of the AOP algorithm, we must point out that it incurs three drawbacks, which are unappealing in practice. First, the formulation in Eq (21) is only suitable for mixed Gaussian impulse noise, i.e. it produces a sub-optimal solution when the observed image is corrupted by pure impulse noise. (ii) Secondly, AOP is a multiple-stage algorithm. Since the minimization sub-problem over u ² needs to be solved exactly in each stage, the algorithm may suffer from slow convergence. (iii) As a by-product of (i), AOP inevitably introduces an additional parameter (that specifies the Gaussian noise level), which is not necessarily readily available in practical impulse denoising problems.

In contrast, our proposed ℓ_0TV method is free from these problems. Specifically, (i) as have been analyzed in Section 2, i.e. our ℓ_0 -norm model is optimal for impulse noise. Thus, our method is expected to produce higher quality image restorations, as seen in our results. (ii) Secondly, we have integrated ℓ_0 -norm minimization into a unified proximal ADMM optimization framework, it is thus expected to be faster than the multiple stage approach of AOP. (iii) Lastly, while the optimization problem in Eq (21) contains two parameters, our model only contains one single parameter.

2.4. Connection with other ℓ_0 -norm optimization techniques

Actually, the optimization technique for the ℓ_0 -norm regularization problem is the key to handling impulse noise. Existing methods such as ℓ_p -norm approximation, the smoothing method [10, 11], the Smoothly Clipped Absolute Deviation (SCAD) penalty method[14], the Minimax Concave Plus

²It actually reduces to the ℓ_2TV optimization problem.

(MCP) penalty method [4] and the reweighted ℓ_1 -norm minimization [1] are not appealing since they only give approximate solutions for the ℓ_0TV problem. In addition, the simple projection gradient descent methods [15] are inapplicable to our model since they assume the objective function is smooth.

Very recently, Lu *et al.* propose a Penalty Decomposition Algorithm (PDA) for solving the ℓ_0 -norm optimization algorithm [7]. As has been remarked in [7], ADMM can also be used for solving ℓ_0TV minimization simply by replacing the quadratic penalty functions in the PDA by augmented Lagrangian functions. Nevertheless, as observed in our preliminary experiments and theirs, the practical performance of their ADMM is worse than that of PDA.

Actually, in our experiments, we found PDA is rather unstable. The penalty function can reach very large values ($\geq 10^8$), and the solution can be degenerate when the minimization problem of the augmented Lagrangian function in each iteration is not exactly solved. This motivates us to design a new ℓ_0 -norm optimization algorithm in this paper. We consider a proximal ADMM algorithm to the MPEC formulation of ℓ_0 -norm since it has a primal-dual interpretation. Extensive experiments have demonstrated that proximal ADMM for solving the “lifting” MPEC formulation for ℓ_0TV produces better image restoration qualities.

3. More Experiments

In this section, we present some additional experimental results to demonstrate the superiority of our proposed ℓ_0TV -PADMM method. Due to page limitations, we were not able to add these results in the submission.

We test the deblurring problem in the presence of impulse noise in our experiments. For ℓ_0TV -AOP, we adapt the author’s image denoising implementation to the image deblurring setting. Since Median Filter Methods (MFM) are not convenient to solve the deblurring problems, we do not test them in here. To generate artificial noisy and blurred images, we blur the original images and then add random-valued noise and salt-and-pepper noise with different densities. We use the following MATLAB scripts to generate a blurring kernel of radius R :

$$\begin{aligned} [\mathbf{x}, \mathbf{y}] &= \text{meshgrid}(-R:R, -R:R) \\ \mathbf{K} &= \text{double}(\mathbf{x}.^2 + \mathbf{y}.^2 \leq R.^2) \\ \mathbf{P} &= \mathbf{K} / \text{sum}(\mathbf{K}(:)) \end{aligned} \quad (22)$$

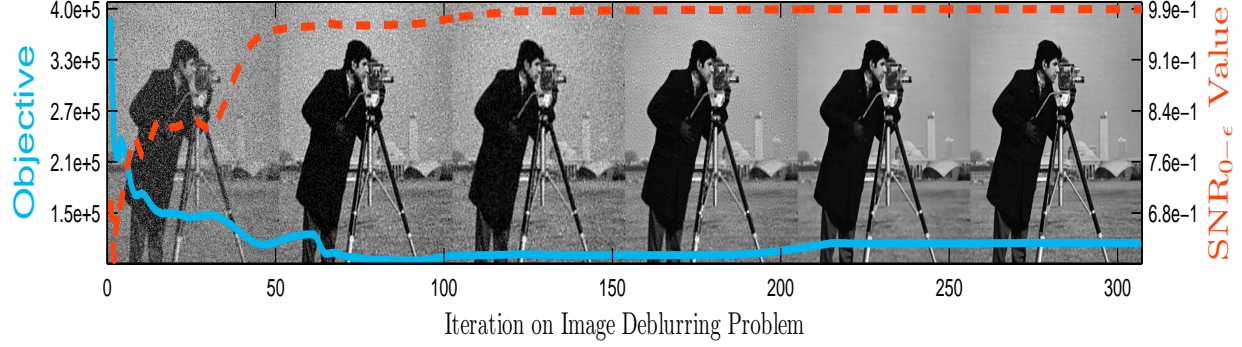


Figure 1: Asymptotic behavior for optimizing Eq (6) to deblur the corrupted ‘cameraman’ image. We plot the value of the objective function (solid blue line) and the SNR value (dashed red line) against the number of optimization iterations. At specific iterations (i.e. 1, 10, 20, 40, 80, and 160), we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

We verify the convergence of ℓ_0TV -PADMM by considering the blurred ‘cameraman’ image subject to 30% random-valued impulse noise. As seen in Figure 1, the asymptotic behavior on image deblurring problems strengthen our conclusions drawn in Section 5.2 in the submission.

3.1. General Image Deblurring Problems

In this subsection, we demonstrate the performance of all methods with their optimal regularization parameters on general deblurring problems. We choose $R = 7$ in Eq(22) in generating the blur kernel. Table 1 shows the recovery results for random-valued impulse noise and salt-and-pepper impulse noise, respectively. We have the following interesting observations. (i) ℓ_0TV -AOP significantly outperforms ℓ_1TV -SBM, and the performance gap becomes larger as the noise level increases. This is because the key assumption in the ℓ_1 model is that $Ku - b$ is sparse at the optimal solution u^* . This does not hold when the noise level is high. (ii) ℓ_0TV -PDA outperforms ℓ_0TV -AOP for high level ($\geq 30\%$) random-valued impulse noise. However, for salt-and-pepper impulse noise, ℓ_0TV -PDA gives worse performance than ℓ_0TV -AOP in most cases. This phenomenon indicates that the Penalty Decomposition Algorithm is not stable for deblurring problems. (iii) By contrast, our ℓ_0TV -PADMM consistently outperforms all methods, especially when the noise level is large. We attribute this result to the “lifting” technique that is used in our optimization algorithm.

3.2. Scratched Image Denoising Problems

In this subsection, we demonstrate the superiority of the proposed ℓ_0TV -PADMM in real-world image restoration problems. Specifically, we corrupt the images with scratches which can be viewed as impulse noise³, see Figure 2. We only consider recovering images using ℓ_0TV -AOP, ℓ_0TV -PDA and ℓ_0TV -PADMM. We show the recovered results in Figure 3. For better visualization of the images recovered by all methods, we also show auxiliary images \mathbf{c} in Figure 4, which show the complement of the absolute residual between the recovered image \mathbf{u} and the corrupted image \mathbf{b} (i.e., $\mathbf{c} = \{1 - |\mathbf{b} - \mathbf{u}|\}$). Note that when c_i is approximately equal to 1, the color of the corresponding pixel at position i in the image is white. A conclusion can be drawn that our method ℓ_0TV -PADMM generates more ‘white’ images \mathbf{c} than the other two methods, since it can identify the ‘right’ outliers in the corrupted image and make the correction using their neighborhood information.

3.3. Colored Image Denoising Problems

Our proposed method can be directly extended to its color version. Since color total variation is not the main theme of this paper, we only provide a basic implementation of it. Specifically, we compute the color total variation channel-by-channel, and take a ℓ_1 -norm of the resulting vectors. Suppose

³Note that this is different from the classical image inpainting problem that assumes the mask is known. In our scratched image denoising problem, we assume the mask is unknown.

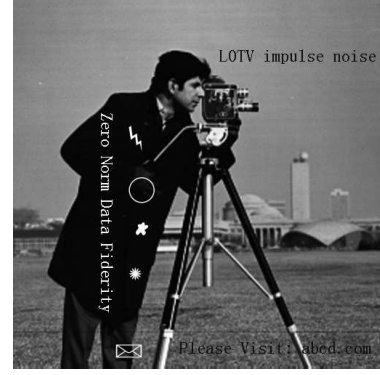
we have RGB channels, then we have the following optimization problem:

$$\min_{\substack{0 \leq u^1 \leq 1 \\ 0 \leq u^2 \leq 1 \\ 0 \leq u^3 \leq 1}} \sum_{k=1}^3 \left(\|\mathbf{o}^k \odot (\mathbf{K}\mathbf{u}^k - \mathbf{b}^k)\|_0 + \lambda \|\nabla \mathbf{u}^k\|_{p,1} \right)$$

where \mathbf{o}^k and \mathbf{u}^k are the prior and the solution of the k th channel. The grayscale proximal ADMM algorithm in Algorithm 1 can be directly extended to solve the optimization above. We demonstrate its applicability in colored image denoising problems in Figure 5. The regularization parameter λ is set to 8 for the three images in our experiments.

References

- [1] E. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14:877–905, 2008.
- [2] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.
- [3] 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.
- [4] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association (JASA)*, 96(456):1348–1360, 2001.
- [5] B. He and X. Yuan. On the $\mathcal{O}(1/n)$ convergence rate of the douglas-rachford alternating direction method. *SIAM Journal on Numerical Analysis (SINUM)*, 50(2):700–709, 2012.
- [6] 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.
- [7] Z. Lu and Y. Zhang. Sparse approximation via penalty decomposition methods. *SIAM Journal on Optimization (SIOPT)*, 23(4):2448–2478, 2013.
- [8] Z. Wen, C. Yang, X. Liu, and S. Marchesini. Alternating direction methods for classical and ptychographic phase retrieval. *Inverse Problems*, 28(11):115010, 2012.
- [9] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Neural Information Processing Systems (NIPS)*, pages 2080–2088, 2009.
- [10] 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.
- [11] L. Xu, S. Zheng, and J. Jia. Unnatural ℓ_0 sparse representation for natural image deblurring. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [12] 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.
- [13] 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.
- [14] C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.
- [15] 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.



(a) SNR_0 value



(b) SNR_1 value



(c) SNR_2 value

Figure 2: Scratched Image Denoising Problems.

Table 1: General Deblurring Problems. The results separated by ‘/’ are SNR_0 , SNR_1 and SNR_2 , respectively.

Alg. Img.	Corrupted	$\ell_1 TV-SBM$	TSM	$\ell_{02} TV-AOP$	$\ell_0 TV-PDA$	$\ell_0 TV-PADMM$
Random-Valued Impulse Noise						
walkbridge+10%	0.63/2.86/3.44	0.72/4.61/8.23	0.72/4.59/8.22	0.81/5.62/10.10	0.75/5.00/8.98	0.92/7.23/13.71
walkbridge+30%	0.52/1.07/-0.00	0.67/4.09/7.40	0.61/3.66/6.80	0.79/5.42/9.71	0.74/4.84/8.68	0.89/6.74/12.48
walkbridge+50%	0.42/-0.19/-1.87	0.58/3.23/5.84	0.46/2.43/4.67	0.75/4.93/8.42	0.73/4.66/8.34	0.86/6.34/11.66
walkbridge+70%	0.31/-1.18/-3.20	0.46/1.97/3.50	0.33/1.11/2.31	0.65/3.22/4.70	0.69/4.31/7.71	0.80/5.59/10.07
walkbridge+90%	0.21/-1.97/-4.19	0.32/0.63/1.17	0.25/0.15/0.50	0.33/0.44/0.62	0.42/1.60/2.82	0.64/3.37/5.47
pepper+10%	0.81/4.92/4.49	0.91/8.34/13.04	0.93/8.35/13.64	0.96/9.72/15.86	0.94/9.02/14.68	0.99/11.12/19.78
pepper+30%	0.66/2.09/0.30	0.83/6.82/10.58	0.83/5.72/9.99	0.96/9.70/15.90	0.93/8.77/14.13	0.99/10.81/19.27
pepper+50%	0.52/0.40/-1.78	0.71/4.61/6.76	0.58/3.38/5.94	0.95/9.27/14.20	0.92/8.49/13.46	0.98/10.31/17.73
pepper+70%	0.38/-0.82/-3.20	0.52/2.34/3.40	0.36/1.61/2.87	0.82/4.93/5.20	0.90/7.83/12.23	0.97/9.88/16.55
pepper+90%	0.23/-1.77/-4.26	0.26/0.61/0.88	0.25/0.42/0.73	0.38/0.89/0.87	0.53/2.45/3.53	0.86/6.33/7.75
mandrill+10%	0.59/1.58/1.27	0.67/3.02/4.86	0.65/2.71/4.31	0.68/2.97/4.56	0.68/3.08/4.96	0.80/4.56/7.93
mandrill+30%	0.49/0.03/-1.71	0.66/2.83/4.56	0.60/2.30/3.84	0.69/3.05/4.68	0.67/2.99/4.80	0.77/4.23/7.25
mandrill+50%	0.40/-1.10/-3.43	0.61/2.43/4.01	0.50/1.65/2.94	0.68/2.92/4.43	0.67/2.90/4.65	0.75/3.91/6.56
mandrill+70%	0.30/-1.99/-4.66	0.48/1.54/2.69	0.41/0.90/1.75	0.65/2.54/3.63	0.65/2.75/4.37	0.72/3.52/5.75
mandrill+90%	0.21/-2.73/-5.64	0.39/0.50/0.86	0.35/0.25/0.50	0.42/0.59/0.73	0.49/1.45/2.46	0.65/2.64/4.02
lenna+10%	0.80/4.15/3.67	0.91/7.52/11.82	0.91/7.19/11.87	0.95/8.51/14.31	0.93/7.81/12.75	0.99/9.74/17.81
lenna+30%	0.66/1.54/-0.28	0.87/6.66/10.55	0.82/5.21/9.15	0.94/8.28/13.71	0.92/7.60/12.31	0.99/9.48/17.22
lenna+50%	0.51/-0.11/-2.38	0.73/4.66/7.46	0.61/3.12/5.65	0.93/7.94/12.62	0.91/7.38/11.79	0.97/9.12/15.96
lenna+70%	0.37/-1.30/-3.78	0.56/2.46/3.73	0.44/1.53/2.92	0.85/5.07/5.77	0.89/6.91/10.90	0.95/8.55/14.61
lenna+90%	0.23/-2.22/-4.82	0.42/0.76/1.08	0.34/0.45/0.79	0.45/0.94/0.91	0.58/2.51/3.90	0.85/5.59/7.20
lake+10%	0.71/4.74/4.91	0.81/7.21/10.91	0.83/7.23/11.26	0.90/8.66/13.84	0.84/7.68/12.05	0.97/9.97/17.85
lake+30%	0.59/2.57/1.26	0.69/5.79/9.28	0.65/5.20/8.87	0.89/8.46/13.31	0.83/7.46/11.60	0.96/9.63/17.07
lake+50%	0.46/1.08/-0.76	0.42/3.58/6.18	0.35/3.13/5.57	0.86/7.90/11.94	0.82/7.19/11.08	0.92/9.08/15.10
lake+70%	0.34/0.02/-2.07	0.19/1.74/3.12	0.22/1.60/2.93	0.66/4.38/5.45	0.79/6.73/10.20	0.89/8.48/13.61
lake+90%	0.22/-0.85/-3.11	0.11/0.63/1.12	0.15/0.44/0.83	0.21/0.62/0.76	0.31/2.21/3.60	0.73/5.45/7.02
jetplane+10%	0.76/3.29/2.13	0.86/6.27/9.17	0.88/6.12/9.67	0.93/7.96/12.61	0.89/6.81/10.48	0.98/9.15/16.43
jetplane+30%	0.63/0.70/-1.80	0.82/5.44/7.55	0.69/3.29/6.48	0.93/7.79/12.07	0.88/6.59/9.99	0.98/8.77/15.86
jetplane+50%	0.49/-0.95/-3.90	0.77/4.32/6.24	0.34/0.89/2.59	0.91/7.01/8.95	0.87/6.32/9.47	0.95/8.35/13.97
jetplane+70%	0.36/-2.13/-5.27	0.33/1.01/2.54	0.21/-0.75/-0.37	0.63/1.34/1.75	0.84/5.78/8.51	0.93/7.67/12.33
jetplane+90%	0.22/-3.05/-6.31	0.11/-0.80/-0.49	0.15/-1.89/-2.53	0.21/-1.73/-2.56	0.30/-0.03/0.52	0.80/4.55/5.26
blonde+10%	0.80/3.49/2.75	0.87/5.57/7.79	0.88/5.71/8.82	0.90/6.17/9.36	0.90/6.34/9.93	0.97/7.43/13.53
blonde+30%	0.66/1.00/-1.17	0.88/5.81/8.44	0.83/4.43/7.44	0.90/6.26/9.48	0.90/6.22/9.64	0.95/7.29/12.57
blonde+50%	0.51/-0.56/-3.18	0.85/5.09/7.09	0.62/2.74/4.94	0.90/6.18/9.13	0.89/6.08/9.34	0.93/6.98/11.46
blonde+70%	0.37/-1.73/-4.59	0.67/3.02/4.50	0.42/1.26/2.53	0.86/4.88/6.01	0.88/5.81/8.80	0.92/6.56/10.62
blonde+90%	0.23/-2.64/-5.63	0.37/0.77/1.32	0.30/0.22/0.65	0.42/0.77/0.91	0.62/2.54/3.93	0.85/5.01/6.30
cameraman+10%	0.78/5.03/4.83	0.84/7.14/10.46	0.89/7.87/12.20	0.94/10.10/15.92	0.90/8.65/13.00	0.99/11.14/19.67
cameraman+30%	0.64/2.39/1.05	0.69/4.54/6.41	0.74/5.26/8.84	0.94/9.99/15.74	0.90/8.41/12.47	0.97/10.83/18.41
cameraman+50%	0.50/0.75/-0.96	0.67/3.49/4.23	0.56/3.09/5.31	0.91/8.46/11.52	0.89/8.12/11.92	0.96/10.45/17.27
cameraman+70%	0.36/-0.45/-2.36	0.60/2.30/2.40	0.37/1.57/2.50	0.72/3.61/3.51	0.86/7.48/10.75	0.94/9.75/15.28
cameraman+90%	0.22/-1.38/-3.40	0.38/1.05/0.98	0.26/0.58/0.70	0.38/0.87/0.75	0.53/2.14/2.71	0.78/4.94/5.20
barbara+10%	0.69/3.62/3.84	0.77/5.65/9.13	0.79/5.66/9.20	0.83/6.47/10.22	0.81/6.05/9.78	0.90/7.61/12.63
barbara+30%	0.57/1.54/0.21	0.73/5.02/8.42	0.67/4.31/7.61	0.83/6.31/9.95	0.80/5.92/9.57	0.88/7.28/11.92
barbara+50%	0.45/0.12/-1.76	0.55/3.40/5.94	0.47/2.69/4.94	0.81/6.07/9.63	0.79/5.77/9.34	0.85/6.79/10.99
barbara+70%	0.34/-0.93/-3.09	0.43/1.89/3.22	0.34/1.31/2.55	0.68/3.54/4.49	0.77/5.45/8.84	0.84/6.45/10.31
barbara+90%	0.22/-1.78/-4.12	0.30/0.55/0.90	0.26/0.33/0.62	0.34/0.61/0.65	0.43/1.91/3.19	0.72/4.39/6.14
boat+10%	0.74/3.88/3.83	0.85/6.54/10.33	0.85/6.33/10.23	0.91/7.79/12.82	0.87/6.79/11.08	0.98/8.92/16.56
boat+30%	0.61/1.55/0.09	0.73/5.19/8.54	0.74/5.00/8.15	0.90/7.60/12.25	0.86/6.61/10.70	0.97/8.53/15.93
boat+50%	0.48/0.00/-1.95	0.67/3.96/6.16	0.51/2.61/5.10	0.87/6.99/10.90	0.84/6.39/10.25	0.93/8.15/14.11
boat+70%	0.35/-1.12/-3.31	0.60/2.55/3.70	0.32/1.17/2.64	0.77/4.25/5.32	0.82/6.00/9.51	0.91/7.57/12.68
boat+90%	0.22/-2.01/-4.35	0.35/0.76/1.31	0.22/0.09/0.67	0.34/0.47/0.75	0.50/2.04/3.41	0.79/5.30/7.51
pirate+10%	0.68/4.06/4.50	0.65/4.87/8.13	0.79/6.25/10.14	0.88/7.79/13.13	0.82/6.75/11.11	0.95/9.18/16.75
pirate+30%	0.56/2.00/0.91	0.65/4.84/7.99	0.61/4.56/7.87	0.87/7.58/12.67	0.81/6.52/10.67	0.93/8.80/15.53
pirate+50%	0.45/0.61/-1.02	0.49/3.13/5.16	0.43/2.78/4.72	0.83/6.64/9.73	0.79/6.28/10.25	0.91/8.42/14.66
pirate+70%	0.33/-0.46/-2.38	0.35/1.53/2.32	0.30/1.33/2.13	0.59/2.90/3.28	0.75/5.73/9.23	0.87/7.60/12.77
pirate+90%	0.21/-1.28/-3.37	0.24/0.43/0.41	0.23/0.34/0.28	0.29/0.50/0.23	0.38/1.56/2.13	0.66/3.58/4.64
Salt-and-Pepper Impulse Noise						
walkbridge+10%	0.61/2.00/0.88	0.72/4.62/8.25	0.80/5.61/10.08	0.81/5.61/10.09	0.76/5.03/9.03	0.94/7.47/14.36
walkbridge+30%	0.48/-0.54/-3.25	0.69/4.25/7.63	0.79/5.40/9.68	0.79/5.40/9.68	0.75/4.90/8.79	0.92/7.19/13.73
walkbridge+50%	0.35/-2.12/-5.31	0.62/3.63/6.52	0.77/5.18/9.27	0.77/5.15/9.22	0.73/4.75/8.52	0.90/6.84/12.92
walkbridge+70%	0.21/-3.25/-6.67	0.53/2.68/4.70	0.75/4.95/8.81	0.75/4.94/8.78	0.71/4.58/8.12	0.86/6.35/11.77
walkbridge+90%	0.08/-4.17/-7.73	0.39/1.11/1.70	0.73/4.68/8.31	0.73/4.66/8.26	0.60/3.52/6.41	0.79/5.42/9.85
pepper+10%	0.80/3.59/1.32	0.92/8.60/13.53	0.96/9.67/15.85	0.96/9.67/15.85	0.94/9.10/14.82	0.99/11.43/20.30
pepper+30%	0.62/0.15/-3.20	0.87/7.47/11.65	0.96/9.55/15.60	0.96/9.55/15.60	0.94/8.92/14.42	0.99/11.21/19.76
pepper+50%	0.45/-1.74/-5.37	0.77/5.64/8.59	0.95/9.46/15.36	0.95/9.47/15.39	0.93/8.68/13.89	0.99/10.81/19.12
pepper+70%	0.28/-3.05/-6.80	0.63/3.50/5.14	0.95/9.09/14.66	0.95/9.08/14.65	0.91/8.32/13.15	0.98/10.17/18.09
pepper+90%	0.11/-4.04/-7.86	0.27/0.51/0.68	0.94/8.81/14.07	0.94/8.77/13.90	0.81/5.79/9.18	0.96/9.43/15.99
mandrill+10%	0.58/0.68/-1.29	0.67/3.03/4.86	0.67/2.91/4.51	0.67/2.90/4.46	0.68/3.09/4.97	0.86/5.26/9.65
mandrill+30%	0.45/-1.69/-5.17	0.66/2.87/4.60	0.67/2.88/4.46	0.67/2.90/4.43	0.68/3.03/4.86	0.83/4.90/8.89
mandrill+50%	0.32/-3.22/-7.19	0.63/2.62/4.21	0.66/2.85/4.44	0.67/2.85/4.40	0.67/2.95/4.73	0.80/4.46/7.85
mandrill+70%	0.19/-4.34/-8.55	0.55/1.96/3.30	0.66/2.82/4.41	0.66/2.82/4.41	0.66/2.84/4.54	0.75/3.95/6.68
mandrill+90%	0.07/-5.24/-9.59	0.38/0.44/0.74	0.65/2.72/4.26	0.65/2.73/4.26	0.60/2.38/3.93	0.70/3.26/5.28
lenna+10%	0.79/2.85/0.49	0.91/7.60/11.97	0.95/8.53/14.39	0.95/8.51/14.35	0.93/7.92/12.87	0.99/9.92/18.38
lenna+30%	0.61/-0.45/-3.93	0.89/6.95/10.92	0.95/8.34/13.95	0.94/8.31/13.86	0.92/7.72/12.51	0.99/9.74/17.99
lenna+50%	0.44/-2.35/-6.13	0.80/5.43/8.51	0.93/7.92/12.93	0.93/7.93/12.77	0.91/7.54/12.10	0.99/9.51/17.34
lenna+70%	0.26/-3.64/-7.55	0.63/3.30/5.09	0.93/7.86/12.90	0.93/7.84/12.85	0.90/7.25/11.51	0.98/9.08/16.37
lenna+90%	0.09/-4.64/-8.64	0.43/0.71/0.74	0.91/7.02/10.91	0.91/7.20/10.93	0.80/5.23/8.47	0.95/8.33/13.97
lake+10%	0.69/3.86/2.43	0.82/7.25/10.98	0.90/8.65/13.81	0.90/8.66/13.82	0.85/7.74/12.13	0.98/10.32/18.46
lake+30%	0.54/1.04/-1.79	0.74/6.15/9.70	0.88/8.44/13.39	0.88/8.44/13.39	0.84/7.56/11.79	0.97/10.08/17.87
lake+50%	0.39/-0.66/-3.89	0.51/4.19/6.88	0.87/8.20/12.90	0.87/8.19/12.88	0.83/7.37/11.39	0.96/9.71/17.04
lake+70%	0.23/-1.87/-5.29	0.23/1.96/3.35	0.86/7.88/12.15	0.86/7.87/12.13	0.81/7.07/10.79	0.93/9.27/15.86
lake+90%	0.08/-2.82/-6.35	0.09/0.16/0.19	0.84/7.57/11.48	0.84/7.53/11.37	0.64/5.12/8.25	0.88/8.16/13.37
jetplane+10%	0.75/2.30/-0.47	0.87/6.43/9.38	0.93/7.97/12.63	0.93/7.96/12.62	0.89/6.90/10.56	0.99/9.56/17.73
jetplane+30%	0.58/-0.89/-4.88	0.81/5.25/7.15	0.93/7.71/11.99	0.93/7.65/11.82	0.88/6.72/10.21	0.99/9.28/17.15
jetplane+50%	0.42/-2.66/-6.95	0.76/4.19/5.23	0.91/7.44/11.39	0.91/7.40/11.27	0.87/6.50/9.74	0.98/9.20/16.14
jetplane+70%	0.25/-3.95/-8.38	0.71/2.95/2.84	0.90/7.03/10.50	0.90/7.02/10.23	0.86/6.17/9.10	0.96/8.39/14.77
jetplane+90%	0.08/-4.93/-9.45	0.63/1.45/0.81	0.89/6.73/9.85	0.89/6.75/9.95	0.72/3.59/6.16	0.92/7.39/11.79
blonde+10%	0.78/2.15/-0.59	0.89/6.22/9.45	0.89/6.01/9.11	0.89/6.02/9.12	0.90/6.39/9.96	0.98/7.93/13.34
blonde+30%	0.61/-1.05/-4.94	0.88/5.86/8.54	0.89/6.09/9.24	0.90/6.09/9.22	0.90/6.29/9.77	0.97/7.73/13.76
blonde+50%	0.44/-2.87/-7.07	0.87/5.40/7.51	0.89/6.05/9.17	0.90/6.08/9.18	0.89/6.18/9.52	0.96/7.43/13.02
blonde+70%	0.27/-4.17/-8.51	0.76				



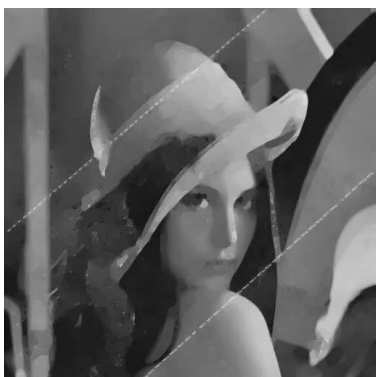
(a) Recovered by $\ell_{02}TV$ -AOP



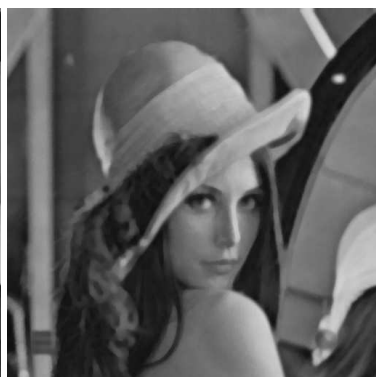
(b) Recovered by ℓ_0TV -PDA



(c) Recovered by ℓ_0TV -PADMM



(d) Recovered by $\ell_{02}TV$ -AOP



(e) Recovered by ℓ_0TV -PDA



(f) Recovered by ℓ_0TV -PADMM



(g) Recovered by $\ell_{02}TV$ -AOP

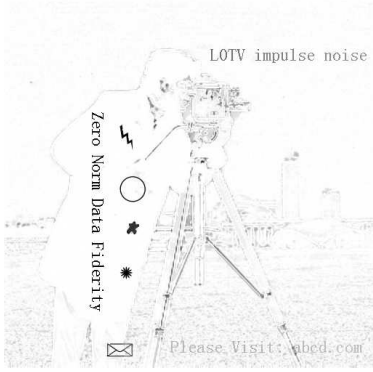


(h) Recovered by ℓ_0TV -PDA

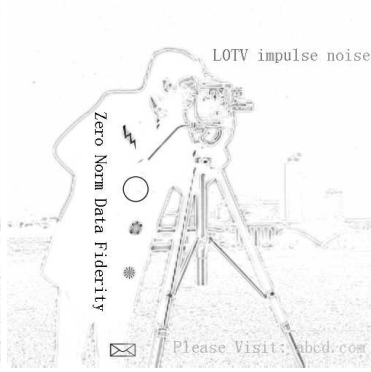


(i) Recovered by ℓ_0TV -PADMM

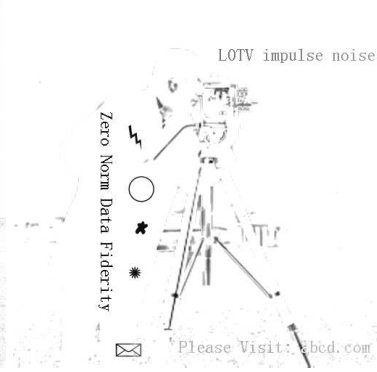
Figure 3: Scratched Image Denoising Problems.



(a) Recovered by ℓ_0TV -AOP



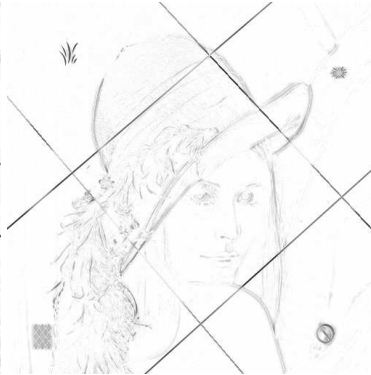
(b) Recovered by ℓ_0TV -PDA



(c) Recovered by ℓ_0TV -PADMM



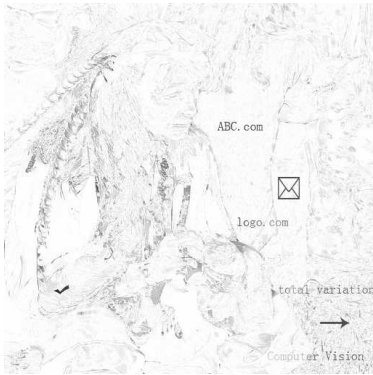
(d) Recovered by ℓ_0TV -AOP



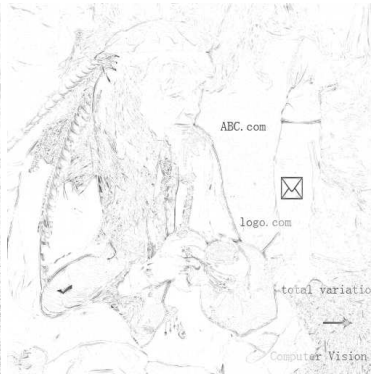
(e) Recovered by ℓ_0TV -PDA



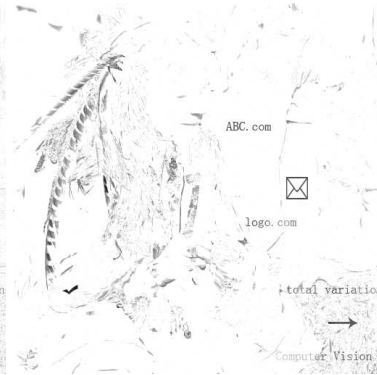
(f) Recovered by ℓ_0TV -PADMM



(g) Recovered by ℓ_0TV -AOP



(h) Recovered by ℓ_0TV -PDA



(i) Recovered by ℓ_0TV -PADMM

Figure 4: Scratched Image Denoising Problems.



(a) clean color 'pepper' image.



(b) corrupted 'pepper' image. $SNR_0 = 0.75$, $SNR_1 = 3.06$, $SNR_2 = 1.95$.



(c) recovered 'pepper' image. $SNR_0 = 0.95$, $SNR_1 = 8.00$, $SNR_2 = 14.01$.



(d) clean color 'lenna' image.



(e) corrupted 'lenna' image. $SNR_0 = 0.75$, $SNR_1 = 3.30$, $SNR_2 = 1.15$.



(f) recovered 'lenna' image. $SNR_0 = 0.97$, $SNR_1 = 12.77$, $SNR_2 = 16.44$.



(g) clean color 'jetplane' image.



(h) corrupted 'jetplane' image. $SNR_0 = 0.74$, $SNR_1 = -0.27$, $SNR_2 = -2.42$.



(i) recovered 'jetplane' image. $SNR_0 = 0.89$, $SNR_1 = 3.21$, $SNR_2 = 7.49$.

Figure 5: Colored Image Denoising Problems.