ℓ_0TV: A New Method for Image Restoration in the Presence of Impulse Noise (Supplementary Material)

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

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 are 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

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

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, $\mathbf{v}_i = 1 - sign(|\mathbf{w}_i|)$. 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:

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

□

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 $\ell_0 TV$ optimization problem. For simplicity, we define:

$$\Delta \triangleq \{z \mid \mathbf{0} \leq \mathbf{z} \leq \mathbf{1}\}. \tag{2}$$

Based on the augmented Lagrangian function of the $\ell_0 TV$ optimization problem, we naturally derive the following KKT conditions of the optimization problem for $\{\mathbf{u}^*, \mathbf{v}^*, \mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\xi}^*, \boldsymbol{\zeta}^*, \boldsymbol{\pi}^*\}$:

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

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].

978-1-4673-6964-0/15/$31.00 ©2015 IEEE

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

$$\lim_{k\to\infty} \boldsymbol{\nabla}\mathbf{u}^{k+1} - \mathbf{x}^{k+1} = 0, \tag{4}$$

$$\lim_{k\to\infty} \mathbf{K}\mathbf{u}^{k+1} - \mathbf{b} - \mathbf{y}^{k+1} = 0, \tag{5}$$

$$\lim_{k\to\infty} \mathbf{o} \odot \mathbf{v}^{k+1} \odot |\mathbf{y}^{k+1}| = 0 \tag{6}$$

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

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

which is equivalent to:

$$\lim_{k\to\infty} -\boldsymbol{\zeta}^{k+1} + \boldsymbol{\pi}^{k+1} \odot \mathbf{v}^{k+1} \odot \mathbf{o} \odot \partial\|\mathbf{y}^{k+1}\|_1 = 0 \tag{7}$$

Moreover, we have the following limit:

$$\lim_{k\to\infty} \mathbf{y}^{k+1} - \mathbf{y}^k = 0. \tag{8}$$

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

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

which is equivalent to:

$$\lim_{k\to\infty} -\boldsymbol{\xi}^{k+1} + \partial\lambda\|\mathbf{x}\|_{p,1} = 0 \tag{9}$$

Clearly, we obtain the following limit:

$$\lim_{k\to\infty} \mathbf{x}^{k+1} - \mathbf{x}^k = 0. \tag{10}$$

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

$$\lim_{k\to\infty} \mathbf{v}^{k+1} \in \arg\min_{0\le\mathbf{v}\le\mathbf{1}} \frac{\beta}{2}\|\mathbf{v} \odot \mathbf{o} \odot \mathbf{y}^{k+1}\|^2 - \langle \mathbf{v}, \mathbf{1} - \mathbf{o} \odot \boldsymbol{\pi}^{k+1} \odot |\mathbf{y}^{k+1}|\rangle,$$

which is equivalent to:

$$\lim_{k\to\infty} \langle \boldsymbol{\pi}^{k+1} \odot \mathbf{o} \odot |\mathbf{y}^{k+1}| - \mathbf{1}, \mathbf{v} - \mathbf{v}^{k+1}\rangle \ge 0, \ \forall \mathbf{v} \in \Delta \tag{11}$$

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

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

Take the limit of the equality constraints, we have:

$$\lim_{k\to\infty} \langle \boldsymbol{\nabla}^T\boldsymbol{\xi}^k + \mathbf{K}^T\boldsymbol{\zeta}^k + \mathbf{D}(\mathbf{u}^{k+1} - \mathbf{u}^k), \mathbf{u} - \mathbf{u}^{k+1}\rangle \ge 0, \quad \forall\ \mathbf{u} \in \Delta. \tag{12}$$

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

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

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

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

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

$$\lim_{k\to\infty} \langle \mathbf{u}^{k+1} - \mathbf{u}^*, \boldsymbol{\nabla}^T(\boldsymbol{\xi}^{k+1} - \boldsymbol{\xi}^*) + \mathbf{K}^T(\boldsymbol{\zeta}^{k+1} - \boldsymbol{\zeta}^*)\rangle + \langle \boldsymbol{\xi}^{k+1} - \boldsymbol{\xi}^*, \boldsymbol{\nabla}(\mathbf{u}^{k+1} - \mathbf{u}^*)\rangle + \langle \boldsymbol{\zeta}^{k+1} - \boldsymbol{\zeta}^*, \mathbf{K}(\mathbf{u}^{k+1} - \mathbf{u}^*)\rangle \ge 0$$

which can be simplified as:

$$\lim_{k\to\infty} 2\langle \mathbf{u}^{k+1} - \mathbf{u}^*, \boldsymbol{\nabla}^T(\boldsymbol{\xi}^{k+1} - \boldsymbol{\xi}^*) + \mathbf{K}^T(\boldsymbol{\zeta}^{k+1} - \boldsymbol{\zeta}^*)\rangle \ge 0 \tag{14}$$

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

$$\lim_{k\to\infty} \langle \mathbf{u}^{k+1} - \mathbf{u}^*, \boldsymbol{\nabla}^T\boldsymbol{\xi}^* - \mathbf{K}^T\boldsymbol{\zeta}^*\rangle \ge 0 \tag{15}$$

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

$$\lim_{k\to\infty} \langle \mathbf{u}^{k+1} - \mathbf{u}^*, \mathbf{D}(\mathbf{u}^k - \mathbf{u}^{k+1})\rangle \ge 0 \tag{16}$$

From the Pythagoras relation[1] and the inequality above, as $k \to \infty$ it follows that

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

[1]Pythagoras 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 $i = 0, 1, ..., k$, we have:

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

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

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

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

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

□

2. Discussions on the connection with Existing Work

In this section, we discuss the connection between the proposed method $\ell_0 TV$-PADMM and prior work.

2.1. Connection with convex optimization method $\ell_1 TV$

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 $\ell_1 TV$, which can be formulated as follows:

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

It is generally believed that $\ell_1 TV$ 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 $\ell_0 TV$ 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, $\ell_1 TV$ 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

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

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:

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

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:

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

They further reformulate the problem above into

$$\min_{\mathbf{u},\mathbf{v}} \ \|\mathbf{v}\odot(\mathbf{K}\mathbf{u}-\mathbf{b})\|_2^2+\lambda\ \|\boldsymbol{\nabla}\mathbf{u}\|_{p,1},$$
$$s.t.\ \mathbf{0}\le\mathbf{v}\le\mathbf{1},\ \langle\mathbf{v},\mathbf{1}\rangle\le n-k$$

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 (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 ($\ge 10^8$), 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 $\ell_{02}TV$-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:

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

[2] It actually reduces to the ℓ_2TV optimization problem.

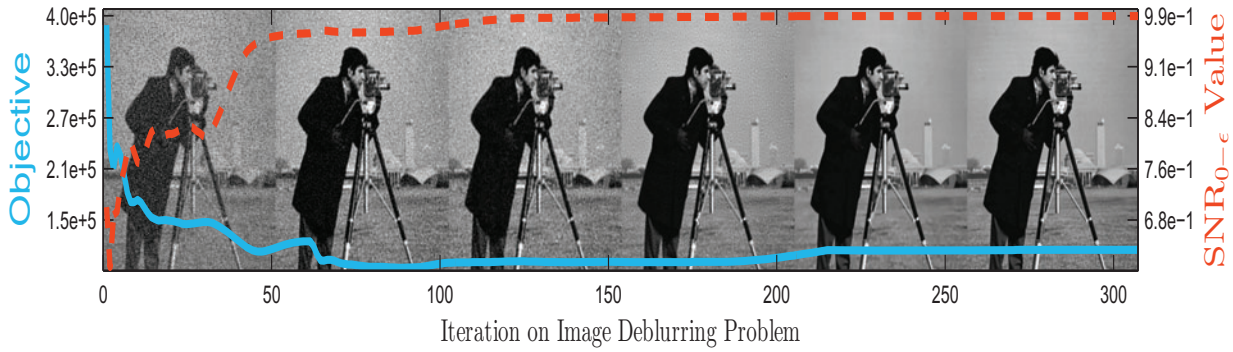


Figure 1: Asymptotic behavior for optimizing Eq (6) to deblur the corrupted 'cameraman' image. We plot the value of the objective function (solid blue line) and the SNR value (dashed red line) against the number of optimization iterations. At specific iterations (i.e. 1, 10, 20, 40, 80, and 160), we 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 inpulse 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**) $\ell_{02}TV$-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 $\ell_{02}TV$-AOP for high level ($\geq 30\%$) random-valued impulse noise. However, for salt-and-pepper impulse noise, ℓ_0TV-PDA gives worse performance than $\ell_{02}TV$-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 noise[3], see Figure 2. We only consider recovering images using $\ell_{02}TV$-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., $\mathbf{c} = \{\mathbf{1} - |\mathbf{b} - \mathbf{u}|\}$). Note that when $\mathbf{c}_i$ 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

[3]Note 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:

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

where $\mathbf{o}^k$ and $\mathbf{u}^k$ are the prior and the solution of the kth 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.

(a) SNR_0 value

(b) SNR_1 value

(c) SNR_2 value

Figure 2: Scratched Image Denoising Problems.

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.

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

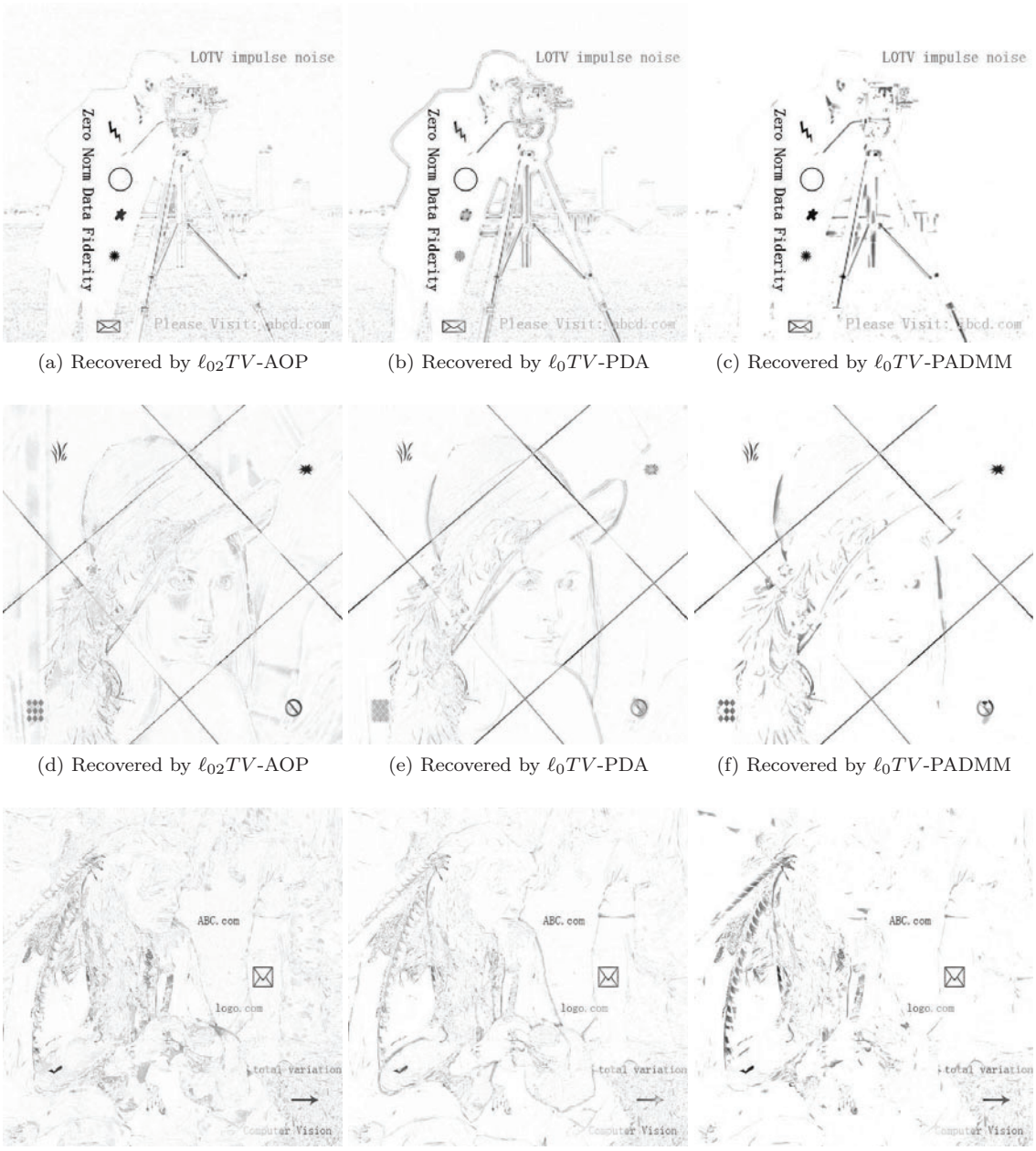
Img. \ Alg.	Corrupted	$\ell_1 TV$-SBM	TSM	$\ell_{02} TV$-AOP	$\ell_0 TV$-PDA	$\ell_0 TV$-$PADMM$
Random-Valued Impulse Noise						
walkbridge+10%	0.63/2.86/3.44	0.72/4.61/8.23	0.72/4.59/8.22	0.81/5.62/10.10	0.75/5.00/8.98	**0.92/7.23/13.71**
walkbridge+30%	0.52/1.07/-0.00	0.67/4.09/7.40	0.61/3.66/6.80	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	**0.92/6.65/10.62**
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	0.56/3.07/5.31	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	0.74/4.50/8.15	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	0.71/4.54/8.12	**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	0.93/7.89/12.87	**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	**0.98/7.93/14.34**
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	0.76/3.88/5.27	0.89/5.96/8.95	0.89/5.96/8.97	0.88/6.00/9.15	**0.94/7.07/12.01**
blonde+90%	0.09/-5.15/-9.58	0.45/1.04/1.38	0.88/5.78/8.71	0.88/5.82/8.82	0.83/4.80/7.41	**0.91/6.34/10.28**
cameraman+10%	0.78/3.95/2.20	0.87/7.81/11.29	0.95/10.15/16.11	0.95/10.14/16.08	0.91/8.74/13.07	**0.99/11.33/20.85**
cameraman+30%	0.63/0.67/-2.15	0.72/4.77/6.59	0.94/9.97/15.68	0.94/9.98/15.69	0.90/8.56/12.70	**0.99/11.09/20.35**
cameraman+50%	0.47/-1.16/-4.26	0.68/3.62/4.47	0.93/9.58/14.75	0.93/9.53/14.58	0.89/8.30/12.20	**0.98/11.20/19.35**
cameraman+70%	0.32/-2.44/-5.66	0.64/2.67/2.56	0.92/9.09/13.60	0.92/9.03/13.41	0.88/8.00/11.52	**0.97/10.34/17.90**
cameraman+90%	0.17/-3.43/-6.73	0.50/1.24/0.56	0.91/8.79/12.90	0.90/8.59/12.37	0.77/5.67/8.57	**0.93/9.33/14.61**
barbara+10%	0.67/2.62/1.08	0.80/5.95/9.47	0.83/6.49/10.29	0.83/6.49/10.28	0.81/6.10/9.84	**0.94/8.02/14.04**
barbara+30%	0.52/-0.15/-3.10	0.76/5.39/8.81	0.83/6.32/10.03	0.82/6.32/10.03	0.81/5.99/9.68	**0.92/7.70/13.24**
barbara+50%	0.37/-1.86/-5.22	0.63/4.07/6.89	0.82/6.16/9.82	0.82/6.17/9.82	0.80/5.88/9.49	**0.89/7.41/12.40**
barbara+70%	0.22/-3.06/-6.62	0.46/2.36/4.01	0.81/6.03/9.61	0.81/6.03/9.60	0.78/5.69/9.20	**0.87/7.01/11.48**
barbara+90%	0.08/-3.99/-7.66	0.29/0.43/0.64	0.80/5.88/9.39	0.80/5.85/9.32	0.65/4.22/7.29	**0.83/6.33/10.23**
boat+10%	0.73/2.76/0.95	0.85/6.57/10.38	0.91/7.80/12.86	0.91/7.80/12.86	0.87/6.86/11.15	**0.98/9.34/17.36**
boat+30%	0.57/-0.28/-3.36	0.79/5.65/9.22	0.90/7.60/12.40	0.90/7.55/12.29	0.86/6.71/10.86	**0.98/9.10/16.77**
boat+50%	0.41/-2.08/-5.50	0.69/4.35/6.85	0.89/7.34/11.84	0.89/7.34/11.83	0.85/6.53/10.46	**0.97/8.81/15.98**
boat+70%	0.25/-3.33/-6.92	0.64/3.09/4.20	0.87/6.99/11.05	0.87/6.99/11.02	0.84/6.30/10.01	**0.95/8.29/14.83**
boat+90%	0.09/-4.31/-8.00	0.53/1.44/1.42	0.86/6.63/10.40	0.85/6.54/10.20	0.73/4.60/7.75	**0.89/7.23/12.15**

(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 $\ell_{02}TV$-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

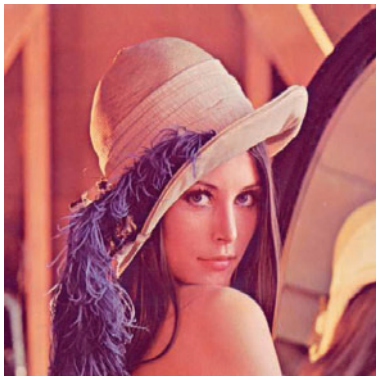
(g) Recovered by $\ell_{02}TV$-AOP (h) Recovered by ℓ_0TV-PDA (i) Recovered by ℓ_0TV-PADMM

Figure 4: Scratched Image Denoising Problems.

(a) clean color 'pepper' image.

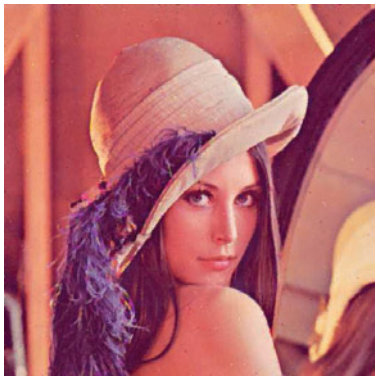
(b) corrupted 'pepper' image. $SNR_0 = 0.75$, $SNR_1 = 3.06$, $SNR_2 = 1.95$.

(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.