Impulse Noise Image Restoration Using Nonconvex Variational Model and Difference of Convex Functions Algorithm

Benxin Zhang™, Guopu Zhu™, Senior Member, IEEE, Zhibin Zhu™, Hongli Zhang™, Member, IEEE, Yicong Zhou™, Senior Member, IEEE, and Sam Kwong™, Fellow, IEEE

Abstract—In this article, the problem of impulse noise image restoration is investigated. A typical way to eliminate impulse noise is to use an $L_1$ norm data fitting term and a total variation (TV) regularization. However, a convex optimization method designed in this way always yields staircase artifacts. In addition, the $L_1$ norm fitting term tends to penalize corrupted and noise-free data equally, and is not robust to impulse noise. In order to seek a solution of high recovery quality, we propose a new variational model that integrates the nonconvex data fitting term and the nonconvex TV regularization. The usage of the nonconvex TV regularizer helps to eliminate the staircase artifacts. Moreover, the nonconvex fidelity term can detect impulse noise effectively in the way that it is enforced when the observed data is slightly corrupted, while is less enforced for the severely corrupted pixels. A novel difference of convex functions algorithm is also developed to solve the variational model. Using the variational method, we prove that the sequence generated by the proposed algorithm converges to a stationary point of the nonconvex objective function. Experimental results show that our proposed algorithm is efficient and compares favorably with state-of-the-art methods.

Index Terms—Difference of convex functions algorithm (DCA), image restoration, impulse noise, nonconvex optimization model.

I. INTRODUCTION

IMAGE restoration is a classical inverse problem and plays an important role in the field of image processing. Specifically, we consider the problem of restoring an image corrupted by impulse noise. Mathematically, the image degradation model can be formulated as $f = N_{imp}(Ku)$, where $f \in \mathbb{R}^m$ is the damaged image, $N_{imp}$ means that the degradation is caused by impulse noise, $K \in \mathbb{R}^{m \times n}$ is an operator, such as convolution and wavelet transform, and $u \in \mathbb{R}^n$ is the original image. Then, our main aim is to recover the original image from the damaged image.

Impulse noise often emerges due to transmission errors or faulty memory locations of hardware. It will damage only a part of the pixels of original image while not affecting the other image pixels. A popular denoising method is the median filter [1], which has high computational efficiency and favorable denoising power. However, the edge information of restored images is lost after median filtering. In order to preserve the edges, variational approaches have been proposed as an important class of image restoration methods, by which the original image $u$ is recovered by minimizing the energy function. Generally, the minimization function of the minimization problem consists of the data fidelity term and regularization term.

The data fidelity term is given by the noise type. According to the statistical property of noise, we can derive the corresponding form. For Gaussian noise, the data fidelity function is usually $L_2$ norm [2]. For non-Gaussian noise, $L_1$ norm is suitable. It is well known that for impulse noise, the fidelity term is $L_1$ norm [4]. But, $L_1$ norm yields biased estimators in statistics. Especially, for high-level impulse noise, the data fidelity term with $L_1$ norm performs poorly. Gu et al. [5] introduced a smoothly clipped absolute deviation (SCAD) function for data fitting term, which processes the desirable oracle property. Some other nonconvex fidelity terms have been discussed in [6], [7], and [8], including the exponential type, Geman function, log penalty, and $L_0$ norm. These nonconvex terms are particularly suitable for restoring the images corrupted with high-level impulse noise.

The regularization term is related to the priori knowledge of the image, such as textures and edges, which are important information and structures. A general form of this term is a composition of the potential function and gradient operator. The classical one is Tikhonov regularization [9], of which the
function is quadratic and smooth. It can be easily minimized using a smooth optimization method. However, this regularization function often eliminates edges and texture details. To deal with this shortcoming, total variation (TV) regularization was proposed in [2] and has been proved very successful. TV term is composed of the $L_1$ norm and gradient operator. It captures the gradient sparsity of an image and has the edge-preserving ability. Also, some variants of TV have been studied in the past twenty years. Recently, the Potts model [10], which adopts $L_0$ norm to ensure sparseness, has received much attention. However, the global optimal solution of the model is difficult to obtain. So, it is expected to find new penalties that have the advantages of $L_1$ and $L_0$ norm while avoiding their disadvantages. There are many choices for nonconvex penalties, for example, capped-L1 [11], the minimax concave penalty (MCP) [12], and Log-norm penalty [13]. These regularization functions were introduced to enhance gradient sparsity. Theoretical analysis and experimental results indicate that nonconvex regularization can obtain a better performance than convex regularization.

By using TV and $L_1$ norm data fitting terms, the TVL1 model is proposed for impulse noise removal [3]. Some efficient algorithms, including the alternating direction method of multipliers (ADMMs) [14] and the primal–dual method [15], have been designed for solving the TVL1 optimization problem. Nikolova et al. [4] proposed a new model that combines $L_1$ norm data fitting and nonconvex regularizer. The nonconvex regularizer offers more possibilities to recover high-quality image. By adopting TV as the regularization term and SCAD as the data fitting function, Gu et al. introduced a new model TVSCAD in [5]. This nonconvex model can achieve higher performance than TVL1, and a difference of convex functions algorithm (DCA) was designed to solve it. Zhang et al. [7] proposed a simple optimization model that makes use of a nonconvex log function for data fitting term and TV regularization. Compared to TVSCAD, this model is easy to choose the scalar parameter. They also developed a new DCA with an adaptive proximal parameter. Zhang et al. [6] introduced some nonconvex-TV models with a nonconvex potential function, and gave a proximal linearized minimization algorithm. Yuan and Ghanem [8] proposed a sparse optimization method, which applies TV regularization and $L_0$ norm data fidelity for impulse noise removal. To solve the nonconvex optimization model, the authors reformulated the problem as a mathematical programming with equilibrium constraints. Very recently, Cui and Fan [16] proposed a nonconvex regularization term plus nonconvex data fitting term model. Further, the authors designed an alternating direction minimization method to solve the optimization problem.

In this article, the issue of impulse noise image restoration is investigated. To solve this issue, a new nonconvex variational model is developed considering that the nonconvex plus nonconvex model may be more suitable for reducing artifacts and eliminating impulse noise. Generally, it is a challenging task to find an efficient algorithm to solve the optimization problem related to nonconvex variational model. To address this, a novel DCA is presented. The main contributions of this work are as follows:

First, a new variational model, which is nonconvex in terms of both TV term and data fidelity term, is proposed for the removal of impulse noise. In the proposed model, the nonconvex TV eliminates artifacts while preserving the sharpness and smoothness of the restored image, meanwhile the nonconvex data fidelity effectively removes the impulse noise.

Second, a novel DCA with an $L_1$ norm proximal term is presented to solve the proposed nonsmooth and nonconvex variational model. To the best of our knowledge, $L_1$ proximal technique has not yet been used in DC programming. With the help of the $\eta$ stationary point, the convergence of the proposed algorithm is also proved. The convergence analysis extends the existing results of general DC programming, thus also makes a theoretical contribution to this kind of issues.

Third, our new model can deal with different image processing tasks, such as image deblurring, image recovery, and image denoising (note that the noise types include salt-and-pepper (SP) impulse noise, random-valued (RV) impulse noise, Gaussian plus SP and RV impulse noise, and Gaussian mixture noise). Experimental results show that the proposed algorithm outperforms other competitors for suppressing impulse noise.

The remainder is organized as follows. In Section II, the new nonconvex model is shown. In Section III, a novel DCA is proposed to deal with this optimization model. The convergence of DCA is established in Section IV. In Section V, experimental results for the proposed model and algorithm demonstrate encouraging performance. In Section VI, the conclusion is presented.

II. MOTIVATION AND NEW MODEL

By combining TV regularization and $L_1$ norm penalized data term, the minimization of TVL1 model for impulse noise removal can be written as

$$\min_{u} \|u\|_{TV} + \mu \| Ku - f \|_1$$

(1)

where $\mu > 0$ is a regularization parameter, and $\| \cdot \|_{TV}$ is TV norm, that is, $\| \nabla u \|_1$. Using the finite difference operation, TV can be discretized into isotropic and anisotropic forms

$$\|u\|_{TV_{\nabla}} := \| Du \|_2, \quad \|u\|_{TV_{\text{aniso}}} := \| Du \|_1$$

where $\nabla$ is gradient operator and $D$ denotes a first-order finite difference of $u$ at every pixel. It is known that TVL1 model works reasonably well only for low-level impulse noise, and performs poorly for high-level impulse noise. The reason may be that both the corrupted data and the noise free data are equally penalized in data fitting, leading to significant difficulty in balancing regularization and data fitting. Moreover, the $L_1$ norm penalty is known to yield biased estimators in statistics. The solution of the TVL1 model substantially deviates from both the data acquisition model and the prior model and, thus, is suboptimal. Then, some nonconvex approaches and correction procedures are proposed.

Yuan and Ghanem [8] proposed the following sparse optimization $L0$TV model:

$$\min_{u} \mu \| Ku - f \|_0 + \| \nabla u \|_1$$

(2)
which takes $L_0$ norm as a data fidelity term and TV term as a regularization term. Model (2) is suitable for image restoration under impulse noise. While the problem described by (2) is a combinatorial optimization problem (NP hard), and it is difficult to obtain its optimal solution. In [17], a corrected TVL1 approach was proposed by solving a series equivalence problems of TVL1. Base on the work of Bai et al. [17], Gu et al. [5] introduced the TVSCAD model that includes TV and SCAD function. The TVSCAD model can be described as

\[
\min_u \|u\|_{TV} + \mu \Phi_\gamma (Ku - f)
\]

where $\Phi_\gamma$ is defined as

\[
\Phi_\gamma (y) = \sum_{i = 1}^m \phi_\gamma (y_i), \; y \in \mathbb{R}^m
\]

and

\[
\phi_\gamma (t) = \begin{cases}
|t|, & \text{if } |t| \leq \gamma_1 \\
-\frac{t^2}{2\gamma_2^2} + \frac{\gamma_1}{\gamma_2} t, & \text{if } \gamma_1 < |t| < \gamma_2 \\
\frac{\gamma_1^2}{\gamma_2^2}, & \text{if } |t| \geq \gamma_2
\end{cases}
\]

where $y = [y_1, \ldots, y_m]^T$ denotes the function variable of $\Phi_\gamma (y)$, $t$ is the function variable of $\phi_\gamma (t)$, $| \cdot |$ denotes the absolute value, $\gamma := (\gamma_1, \gamma_2)$ is a pair of parameters, and $\gamma_2 > \gamma_1 > 0$. Then, a DCA is proposed to solve this model. The resulting subproblem is solved by ADMM, and the global convergence is also established. Zhang et al. [6] proposed a nonconvex model, which is defined as

\[
\min_u \mu P (Ku - f) + \|u\|_{TV}
\]

where $P$ is a potential data fitting function, such as the exponential-type (ET) function. Models (3) and (4) can be considered as the relaxation of (2). Such models can only lead to suboptimal estimation since they give approximate solutions of the problem of (2). Cui and Fan [16] introduced a model by adopting the nonconvex data fitting and regularization terms. Their model is expressed as

\[
\min_u \mu P (Ku - f) + Q (\nabla u)
\]

where $P$ and $Q$ are semialgebraic functions, and both of them can be rewritten as the sum of two functions. By the semiconvergence property of denoising problems, the optimal estimation of (5) converges toward the desired solution first, and then diverges from it after a number of iterations.

The data fitting term containing nonconvex functionals has been used for image restoration under impulsive noise. However, TV regularization was used as the regularizer in these works [5], [6], [8]. Hence, the artifacts may still exist in the restored image. Considering the effectiveness of the artifact removal of nonconvex regularization [16] and the robustness of nonconvex data fidelity to outliers, we propose a new model that combines the nonconvex data fitting term and nonconvex regularization term. The model is given as

\[
\min_u \mu \Phi_\gamma (Ku - f) + \Psi_\gamma (\nabla u)
\]

where $\Psi_\gamma$ is defined as

\[
\Psi_\gamma (v) = \sum_{i = 1}^n \psi_\gamma (v_i), \; v \in \mathbb{R}^n
\]

and

\[
\psi_\gamma (t) = \frac{1}{s} \log (1 + s |t|)
\]

where $s$ is a parameter larger than 0, and $v = [v_1, \ldots, v_n]^T$ denotes the function variable of $\Psi_\gamma (v)$. The motivation of using the SCAD function here is to enforce less or even no data fitting and more regularization whenever $(Ku)$ deviates significantly from $f_i$. This is quite reasonable because the $i$th pixel is more likely to be corrupted in such a case. For the $i$th pixel such that $(Ku - f_i)$ is sufficiently small, the absolute penalty is kept. The structure of SCAD can discriminate the difference between the corrupted and uncorrupted pixels and provide good restorations. Moreover, the SCAD and Log penalty functions make the resulting estimator possesses three desired properties, that is, continuity, sparsity, and unbiasedness [5]. Hence, the solution of (6) is optimal and robust.

### III. Algorithm

In this section, we first introduce some preliminaries that will be used in the proposed approach. Then, a novel DCA is presented.

#### A. Some Properties of $\Phi_\gamma$ and $\Psi_\gamma$

We show some properties of $\Phi_\gamma$ and $\Psi_\gamma$, which are important for a constructive algorithm. First, we consider the functions $\nu_\gamma (t)$ and $\phi_\gamma (t)$ that can be induced by $\phi_\gamma$ and $\psi_\gamma$, that is

\[
\nu_\gamma (t) = |t| - \phi_\gamma (t), \; \gamma > 0
\]

\[
\phi_\gamma (t) = |t| - \psi_\gamma (t), \; s > 0.
\]

From [13], $\nu_\gamma (t)$ and $\phi_\gamma (t)$ are continuously differentiable and convex on $R$.

Note that in order to apply the difference of convex functions programming to $\Phi_\gamma$ and $\Psi_\gamma$, $\nu_\gamma (t)$ and $\phi_\gamma (t)$ are defined in (7) and (8), respectively. Let $\nu_\gamma (t)$ be given by (7), then set $\Xi_\gamma : R^m \rightarrow R$ as

\[
\Xi_\gamma (y) = \sum_{i = 1}^m \nu_\gamma (y_i).
\]

Similarly, $\phi_\gamma (t)$ is given by (8). Set $\Theta_\gamma : R^m \rightarrow R$ as

\[
\Theta_\gamma (v) = \sum_{i = 1}^m \phi_\gamma (v_i).
\]

Note that $\Xi_\gamma$ and $\Theta_\gamma$ are continuously differentiable and convex [13].

From (9) and (10), we also can obtain

\[
\Phi_\gamma (y) = \|y\|_1 - \Xi_\gamma (y)
\]

\[
\Psi_\gamma (v) = \|v\|_1 - \Theta_\gamma (v).
\]
Hence, we can rewrite the proposed model (6) as a DC programming.

B. DC Programming and Novel DCA

DC programming is used to minimize a problem which can be written as the difference of convex functions [18], [19]. By (11) and (12), we can decompose SCAD and log functions as the difference of convex functions, that is

\[ \Psi_x(v) = \|\nabla u\|_1 - \Theta_x(\nabla u). \]

By integrating the above two equations into (6), we obtain

\[ \min_u \|\nabla u\|_1 + \mu \|Ku - f\|_1 - (\Theta_x(\nabla u) + \mu \nabla_x (Ku - f)). \] (13)

Let

\[ g(u) = \|\nabla u\|_1 + \mu \|Ku - f\|_1 \]

and

\[ h(u) = \Theta_x(\nabla u) + \mu \nabla_x (Ku - f) \]

then, (13) can be rewritten as the difference of the cost function \( g(u) \) and \( h(u) \)

\[ \min_u F(u) = g(u) - h(u). \] (14)

So, (14) is a typical DC programming.

It is known that the aim of DCA is to solve the DC programming and its dual problem. Generally, DCA optimizes (14) via linearizing \( h(u) \) and solving the following convex problem:

\[ u_{k+1} = \arg \min_u \{g(u) - (h(u_k) + \langle \nabla h(u_k), u - u_k \rangle)\} \] (15)

where \( \langle \cdot, \cdot \rangle \) denotes the inner product between two vectors. Gu et al. [5] added an \( L_2 \) norm proximal term to (15) and their convex problem is

\[ u_{k+1} = \arg \min_u \left\{ g(u) - (h(u_k) + \langle \nabla h(u_k), u - u_k \rangle) + \frac{\eta}{2} \|u - u_k\|^2 \right\} \] (16)

where \( \eta > 0 \). Different from the above scheme, we introduce a novel DCA with the \( L_1 \) norm proximal term as follows:

\[ u_{k+1} = \arg \min_u \{g(u) - (h(u_k) + \langle \nabla h(u_k), u - u_k \rangle) + \eta \|u - u_k\|_1\}. \] (17)

Then, our DCA for solving (6) is described as

\[ u_{k+1} = \arg \min_u \|Du\|_2 + \mu \|Ku - f\|_1 - (\bar{\mu}^T \Theta_x(Du_k) + \mu K^T \nabla_x (Ku_k - f), u - u_k) + \eta \|u - u_k\|_1. \] (18)

Note that the isotropic discretization TV is adopted in this article and the case of anisotropic TV is completely similar. The objective function of (18) is convex and has a global optimal solution.

In order to use the popular ADMM [20], [21], [22] for solving problem (18), we express (18) as an equivalent form

\[ u_{k+1} = \arg \min_u \{\|Du\|_2 - (\nabla \Theta_x(Du_k), Du) + \mu \|Ku - f\|_1 - \mu \nabla_x (Ku_k - f), Du) + \eta \|u - u_k\|_1\}. \] (19)

By introducing free variables \( w, z, d, p, \) and \( q, \) and defining \( w = Du, z = Ku - f, p_k = \nabla \Theta_x(w_k), q_k = \nabla_x (z_k), d = u, \) (19) can be rewritten as

\[ \min_{u,w,z,d} \|w\|_2 - (p_k, w) + \mu (\|z\|_1 - (q_k, z)) + \eta \|d - u_k\|_1 \]

s.t. \( w = Du, z = Ku - f, d = u. \) (20)

The augmented Lagrangian function of (20) is

\[ \mathcal{L}(w, z, d, u, \lambda_w, \lambda_z, \lambda_d) = \|w\|_2 - (p_k, w) - \lambda_w^T (w - Du) + \frac{\beta_w}{2} \|w - Du\|^2 + \mu (\|z\|_1 - (q_k, z)) - \lambda_z^T (z - (Ku - f)) + \frac{\beta_z}{2} \|z - (Ku - f)\|^2 + \eta \|d - u_k\|_1 - \lambda_d^T (d - u) + \frac{\beta_d}{2} \|d - u\|^2 \]

where \( \beta_w, \beta_z, \beta_d > 0 \) are penalty factors, and \( \lambda_w \in \mathbb{R}^{2n}, \lambda_z \in \mathbb{R}^m, \) and \( \lambda_d \in \mathbb{R}^m \) are multipliers. By (20), we have the following ADMM when initial points start at \( u_0 \) and \( \lambda_w^0, \lambda_z^0, \lambda_d^0 \):

\[ \left( u^{j+1}, z^{j+1}, d^{j+1} \right) = \arg \min_{w,z,d} \mathcal{L}(w, z, d, u, \lambda_w, \lambda_z, \lambda_d) \] (21)

\[ u^{j+1} = \arg \min_u \mathcal{L}(w^{j+1}, z^{j+1}, d^{j+1}, u, \lambda_w, \lambda_z, \lambda_d) \] (22)

\[ \begin{aligned}
\lambda_w^{j+1} &= \lambda_w^j - \alpha \beta_w (w^{j+1} - Du^{j+1}) \\
\lambda_z^{j+1} &= \lambda_z^j - \alpha \beta_z (z^{j+1} - (Ku^{j+1} - f)) \\
\lambda_d^{j+1} &= \lambda_d^j - \alpha \beta_d (d^{j+1} - u^{j+1})
\end{aligned} \] (23)

where \( \alpha \in (0, (1 + \sqrt{5})/2). \) In fact, the scheme of (21)–(23) is ADMM for two blocks of variables \( (w, z, d) \) and \( u. \) Hence, the convergence of this ADMM can be guaranteed by the classical results in [23].

Next, we show how to solve the subproblems (21) and (22). In fact, \( w^{j+1}, z^{j+1}, \) and \( d^{j+1} \) are the solutions of proximity operators about \( \|\cdot\|_2 \) and \( \|\cdot\|_1. \) So, the \( w, z, d\)-subproblems in (21) have a closed form as

\[ w^{j+1} = \max_w \left\{ \|Du^{j+1} + \frac{\lambda_w^{j}}{\beta_w} + p_k\|_2 - \frac{1}{\beta_w}, 0 \right\} \] (24)

\[ z^{j+1} = \max_z \left\{ Ku^{j+1} - f - \frac{\lambda_z^{j} + \mu q_k}{\beta_z} - \frac{\mu}{\beta_z}, 0 \right\} \cdot \text{sign} \left( Ku^{j+1} - f - \frac{\lambda_z^{j} + \mu q_k}{\beta_z} \right) \] (25)

\[ d^{j+1} = \max_d \left\{ \|u^{j+1} + \frac{\lambda_d^{j}}{\beta_d} - u_k\|_2 - \frac{\eta}{\beta_d}, 0 \right\} \cdot \text{sign} \left( u^{j+1} + \frac{\lambda_d^{j}}{\beta_d} + u_k \right) \] (26)

where \( \|\cdot\| \) is componentwise absolute value and sign represents the signum function. To address the issue that the divisor is zero and avoid the instability of numerical computation, \( 0 \cdot (0/0) = 0 \) is assumed.
Algorithm 1 DCA for Solving (6)
Set $\mu, \beta_w, \beta_z, \beta_d, \epsilon, \gamma, \gamma_1, \eta > 0$, $u_0, p_0 = \nabla \Theta_x(Du_0)$, $q_0 = \nabla \Theta_x(Ku_0 - f)$, $k_{max}, j_{max}$.
For $k = 0, 1, 2, \ldots, k_{max}$
Given $\lambda^w, \lambda^z, \lambda^d$, and $u_k, p_k, q_k$.
For $j = 0, 1, 2, \ldots, j_{max}$
Calculate $w^{j+1}$ according to (24).
Calculate $z^{j+1}$ according to (25).
Calculate $d^{j+1}$ according to (26).
Calculate $u^{j+1}$ according to (27).
Update $\lambda^w, \lambda^z, \lambda^d$ via (23).
If $\|u^{j+1} - u^j\|_2/(1 + \|u\|_2) < \epsilon$, break.
End
Set $u_{k+1} = u^{j+1}$ and compute $p_{k+1} = \nabla \Theta_x(Du_{k+1})$, $q_{k+1} = \nabla \Theta_x(Ku_{k+1} - f)$.
End

The minimization $u$-subproblem (22) is a least square. The corresponding normal equation is
\[
(D^T D + \frac{\beta_z}{\beta_w}K^T K + \frac{\beta_d}{\beta_w}I)w^{j+1} = D^T \left(w^j - \frac{\lambda^j}{\beta_w}\right) + \frac{\beta_z}{\beta_w}K^T \left(z^j - \frac{\lambda^j}{\beta_z}\right) + \frac{\beta_d}{\beta_w}K^T f + \frac{\beta_d}{\beta_w} \left(d^j - \frac{\lambda^j}{\beta_d}\right).
\]
(27)

When $\beta_w, \beta_z, \beta_d > 0$, the coefficient matrix $D^T D + (\beta_z/\beta_w)K^T K + (\beta_d/\beta_w)I$ is nonsingular. Under the periodic boundary conditions, this matrix can be diagonalized by Fourier transform. So, the solution of normal equation (27) can be given by two fast Fourier transforms. If the matrix has no special structure to use, it can be commonly solved by the conjugate gradient method.

Finally, we show the novel DCA for solving the proposed model (6) in Algorithm 1.

C. Computational Complexity

The computational complexity of the proposed algorithm can be analyzed as follows. In the outer loop, the complexity of calculating Euclidean gradient is $O(n)$, where $n$ is the image size. In the inner loop, the $w, z$ and $d$ subproblems are solved using the shrinkage operators in linear time, so the complexity of each of them is $O(n)$; the computational cost associated with the $u$ subproblem is $O(n \log n)$ if its solution is obtained via fast Fourier transform and inverse transform, and is $O(n^2)$ if its solution is achieved via the conjugate gradient method; the update of the Lagrangian multipliers via (23) can be implemented straightforwardly in $O(n)$ time. Thus, the worst case complexity of the proposed algorithm for solving problem (6) is $O(k_{max}n \log n)$ or $O(k_{max}n^2)$. It should be pointed out that the complexity of our new algorithm is the same as those of the methods of [5], [6], and [16], but is lower than that of the classical ADMM presented in [8], [14], and [23] [i.e., $O(Tn \log n)$ or $O(Tn^2)$ if $k_{max}=T$, where $T$ is the total iteration number of the classical ADMM.

IV. CONVERGENCE

Using the tools from variational analysis, we prove that Algorithm 1 is convergence in this section. First, a simple property of subdifferential is established.

Lemma 1: If $\xi_{k+1} \in \partial \|u_{k+1} - u_k\|_1$, then
\[
\|u_{k+1} - u_k\|_1 \leq \langle \xi_{k+1}, u_{k+1} - u_k \rangle.
\]

Proof: Let $J(u) = \|u - u_k\|_1$. By the definition of subdifferential, we obtain
\[
J(u_k) - J(u_{k+1}) = 0 - \|u_{k+1} - u_k\|_1 \
\geq \langle \partial J(u_{k+1}), u_k - u_{k+1} \rangle \
= \langle \xi_{k+1}, u_k - u_{k+1} \rangle.
\]

Lemma 2: For the sequence generated by Algorithm 1, we have
\[
F(u_k) - F(u_{k+1}) \geq \eta \|u_{k+1} - u_k\|_1
\]
and $\{u_k\}$ is bounded and convergent.

Proof: Because $g(u)$ is a convex function, we know
\[
g(u_k) - g(u_{k+1}) \geq \langle \partial g(u_{k+1}), u_k - u_{k+1} \rangle.
\]

Similarly, we obtain
\[
h(u_{k+1}) - h(u_k) \geq \langle \partial h(u_k), u_{k+1} - u_k \rangle.
\]

It follows from (17) that:
\[
\partial h(u_k) - \eta \partial \|u_{k+1} - u_k\|_1 \in \partial g(u_{k+1}).
\]

Combining (29) and (30), we obtain
\[
F(u_k) - F(u_{k+1}) \geq \langle -\partial g(u_{k+1}) + \partial h(u_k), u_{k+1} - u_k \rangle \
= \langle \eta \xi_{k+1}, u_{k+1} - u_k \rangle \
\geq \eta \|u_{k+1} - u_k\|_1
\]
where (31) and Lemma 1 are used.

For $k = 0, 1, \ldots$, summing all the inequalities in (28), we deduce that
\[
\sum_{k=0}^{\infty} \|u_{k+1} - u_k\|_1 \leq \frac{1}{\eta} F(u_0)
\]
which implies that $\{u_k\}$ is a Cauchy sequence and is convergent. The assertion is proved.

Before we show the convergence result, the definition of $\eta$ stationary point [24] is given as below.

Definition 1: For $\eta > 0$, $u^*$ is $\eta$ stationary point of $F(u)$ if there exists $\xi \in \partial F(u)$ such that $|\xi|_{\infty} \leq \eta$.

Theorem 1: The limit point $u^*$ of $\{u_k\}$ that is generated by Algorithm 1 is an $\eta$ stationary point of the objective function $F(u)$.

Proof: From the relationship (31), we have
\[
\partial F(u_{k+1}) \ni \partial g(u_{k+1}) - \partial h(u_{k+1}) \
= \nabla h(u_k) - \nabla h(u_{k+1}) - \eta \xi_{k+1}.
\]

Authorized licensed use limited to: Universidade de Macau. Downloaded on March 21,2024 at 01:53:16 UTC from IEEE Xplore. Restrictions apply.
Due to the boundedness of $\xi_{k+1} \in [-1, 1]$ and $\{u_k\}$, we can find a common subsequence $\{k_j\}$, such that $\tilde{\xi}_{k_j} \rightarrow \xi^*$ and $u_{k_j} \rightarrow u^*$. Replacing $k$ by $k_j$ in (32), taking the limit and using the continuity of the function $\nabla h(u)$, we obtain $-\eta \xi^* \in \partial F(u^*)$. By $\|\eta \xi^*\|_{\infty} \leq \eta$, $u^*$ is an $\eta$ stationary point.

V. EXPERIMENTAL VALIDATION

We evaluate the proposed model and algorithm on image restoration problem with impulse noise in this section. The experiments include image deblurring, image denoising and image recovery. All code are performed on a PC with 2.9-GHz processor and 16-GB RAM.

A. Image Deblurring

In the first example, we show the effectiveness of the proposed method in suppressing impulse noise and restoring the blurred image. Impulse noise can be classified into two types, namely SP noise and RV noise. If a grayscale image is degraded by SP noise, then some of its pixels will change to the random noise level $r$ ($0 \leq r \leq 1$) is given as

$$f_i = \begin{cases} 0, & \text{with probability } \frac{\xi}{2} \\ 255, & \text{with probability } \frac{\xi}{2} \\ u_i, & \text{with probability } 1 - r \end{cases}$$

where $u$ denotes the original grayscale image, and $i$ denotes the pixel location. Similarly, if a grayscale image is degraded by RV noise, then some of its pixels will change to the random pixel location. Similarly, if a grayscale image is degraded by RV noise, then some of its pixels will change to the random values which are between the minimum and the maximum pixel values. The noisy $f$ corrupted by the RV noise with noise level $r$ ($0 \leq r \leq 1$) is expressed by

$$f_i = \begin{cases} d_i, & \text{with probability } r \\ u_i, & \text{with probability } 1 - r \end{cases}$$

where $d_i$ is uniformly distributed in $[0, 255]$. In general, the high-level impulse noise means that more than 50% pixels of an entire image are corrupted. In this section, the experiment results of image deblurring with SP and RV noise are presented.

1) Compared With TVL1 and TVSCAD: In this part, the new model (6) is compared with the TVL1 model (1) and TVSCAD model (3). For high-level impulse noise, TVL1 model works poorly. In order to overcome the limitation of TVL1, TVSCAD is proposed. Moreover, our proposed model (6) is an improved version of TVSCAD, where the nonconvex log function is instead of the $L_1$ norm in the regularization term of TVSCAD. Moreover, TVSCAD model is solved by the classical DCA with $L_2$ proximal term, while the new model is solved by a novel DCA with $L_1$ proximal term. Hence, our approach is called NNDCL1, which means that nonconvex plus nonconvex model is solved by DC programming with $L_1$ proximal term.

In our experiments, a $9 \times 9$ Gaussian blur with standard deviation 10 and an average blur with size $9 \times 9$ are considered. Also, SP or RV noises will be added after blur the image. As we know, RV noise is more difficult to removal than SP noise since the noise value can be arbitrary numbers between the min and max pixel value. Therefore, we test 50%, 70%, and 90% noise levels for SP noise and 40%, 60%, and 80% noise levels for RV noise. The testing images are House ($256 \times 256$) and Peppers ($512 \times 512$), as shown in Fig. 1. The quality of the restoration image is evaluated by the signal to noise ration (SNR) and the structure similarity (SSIM) [25]. Obviously, the higher values of SNR and SSIM, the better of the restoration image.

The parameters of the tested models should be specified. These parameters have a great influence on the numerical performances. The best choice of the parameter $\mu$ is known to be problem dependent and very hard to find. So, we adjust the parameters one by one for each image. For the regularization parameter $\mu$, we first determine the optimal parameter value $\mu$ of TVSCAD by solving a series of problems (3) as done in [5]. Then, the best $\mu$ values of TVSCAD are also used in our model (6). For a fair comparison, the time consumed in parameter tuning is not taken into account for all the tested methods. For the TVL1 model, $\mu$ takes 1.75 for both the two types of testing noise. For the TVSCAD model and our model, $\mu$ takes different values for the two types of noise. Specifically, under the SP noise, $\mu$ takes 22 for House image; and under the SP noise with noise levels of 50%, 70%, and 90%, it takes 21, 7, and 17, respectively, for Peppers image. While, under the RV noise with noise level of 80%, $\mu$ takes 5 for House and 3.75 for Peppers; under the RV noise with noise levels of 60% and 40%, it takes 5 and 21, respectively, for two test images. In order to make a fair evaluation, the function $F_\eta$ in (3) and (6) will be taken the same for high-level noise. As suggested in [5], we set $\gamma_1 = 0.08/k$ and $\gamma_2 = \max(0.2 \times 0.85^{k-1}, 0.1)$ for SP noise and $\gamma_1 = 0.0001$ and $\gamma_2 = 0.5$ for RV noise, About $x \in \Psi_\eta$, it takes 0.035 for SP noise and 0.095 for RV noise.

Next, we consider these parameters in compared and proposed algorithms. For all the compared algorithms, we manually choose the regularization and the algorithmic parameters that yield the best SNR and SSIM in the experiments. For nonconvex problems, it is impossible to make an optimization algorithm achieve its best performance for all tests using a set of fixed parameters. Hence in general, the parameters of the optimization algorithm are tuned case by case to obtain much better results for each test. The proximal term can make the problem well defined and stabilize the method. It mentions that the proximal parameter $\eta$ in (17) is related to the optimality condition. Hence, we set $\eta = 0.001$ in (16) and (17) throughout. It also noted that both of TVSCAD and NNDCL1 use ADMM to solve the subproblems (16) and (17). Then, we set $\beta_0 = 5$, $\beta_2 = 10$, $\beta_3 = 100$ for SP noise, $\beta_0 = 2$, $\beta_2 = 5$, $\beta_3 = 650$ for RV noise, and $\alpha = 1.618$ in all tests. For the parameters of TVL1 method, we set them as done in [23].
TABLE I
COMPARISON RESULTS OF DIFFERENT ALGORITHMS FOR SP AND RV NOISE

<table>
<thead>
<tr>
<th>Images</th>
<th>Level</th>
<th>Blurr</th>
<th>TVL1</th>
<th>TVSCAD</th>
<th>NNDCL1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>SNR</td>
<td>SSIM</td>
<td>SNR</td>
</tr>
<tr>
<td>50% (SP)</td>
<td>Gaussian</td>
<td>4.2052</td>
<td>11.6934</td>
<td>0.6741</td>
<td>22.8231</td>
</tr>
<tr>
<td>Average</td>
<td>4.1719</td>
<td>11.8543</td>
<td>0.7691</td>
<td>22.0825</td>
<td>15.6402</td>
</tr>
<tr>
<td>70% (SP)</td>
<td>Gaussian</td>
<td>6.0898</td>
<td>9.5297</td>
<td>0.7227</td>
<td>27.6575</td>
</tr>
<tr>
<td>Average</td>
<td>5.9627</td>
<td>9.7914</td>
<td>0.7273</td>
<td>29.3281</td>
<td>14.9913</td>
</tr>
<tr>
<td>90% (SP)</td>
<td>Gaussian</td>
<td>5.5594</td>
<td>5.3728</td>
<td>0.6316</td>
<td>29.8906</td>
</tr>
<tr>
<td>Average</td>
<td>6.2669</td>
<td>5.7759</td>
<td>0.6238</td>
<td>28.4219</td>
<td>12.2683</td>
</tr>
<tr>
<td>House</td>
<td>40% (RV)</td>
<td>Gaussian</td>
<td>4.2344</td>
<td>11.8545</td>
<td>0.7726</td>
</tr>
<tr>
<td>Average</td>
<td>4.2031</td>
<td>12.0024</td>
<td>0.7728</td>
<td>22.0156</td>
<td>15.4914</td>
</tr>
<tr>
<td>60% (RV)</td>
<td>Gaussian</td>
<td>2.3438</td>
<td>9.4238</td>
<td>0.7227</td>
<td>59.0313</td>
</tr>
<tr>
<td>Average</td>
<td>2.6094</td>
<td>9.7586</td>
<td>0.7247</td>
<td>57.9375</td>
<td>12.7952</td>
</tr>
<tr>
<td>80% (RV)</td>
<td>Gaussian</td>
<td>3.9063</td>
<td>3.3515</td>
<td>0.6145</td>
<td>58.9063</td>
</tr>
<tr>
<td>Average</td>
<td>3.8750</td>
<td>3.8104</td>
<td>0.6236</td>
<td>57.5313</td>
<td>10.1306</td>
</tr>
</tbody>
</table>

For initialization of all algorithms, we set $u_0 = f$. To compute $u_{k+1}$ from $u_k$ in (16) and (17), ADMM starts at $u_k$ and the initial multipliers are zeros when launching ADMM as discussed in [5]. ADMM will terminate when $\epsilon \leq 0.0001$ or the inner iteration number is met. The maximum outer iteration of TVSCAD and NNDCL1 is 5 for SP noise and 10 for RV noise for high-level noise. But, the maximum outer iteration is set as 5 for 40% and 50% low density noise. The reason is that the quality has no much more improvement after several iterations.

Now, we show the compared results of images corrupted by Gaussian and average blur with SP and RV noise. The partial visual comparisons are shown in Figs. 2–5, which include the blurry noisy images and the restored images by different methods. It can be seen that both TVSCAD and NNDCL1 outperform TVL1, especially for high-level noise. Comparing the results of TVSCAD and NNDCL1, we see that NNDCL1 performs competitive with TVSCAD and the two methods adequately restore the images from the very high-level noise. The quantitative evaluations of experiment results are reported in Table I, which gives CPU time (seconds), SNR (dB) and SSIM. Note that each algorithm is tested ten times on the same image under the same random noise level, then each result reported in Table I is obtained by averaging the ten test results. From Table I, one can see that the new model obtains higher SNR and SSIM values in most cases and CPU.
times of NNDCL1 and TVSCAD are almost the same. The convergence curves of NNDCL1 and TVSCAD are plotted in Figs. 6 and 7. We can see that the curves of SNR values increase as the outer iteration numbers increase. NNDCL1 performs better than TVSCAD in the case of SP noise. For RV noise, NNDCL1 is also competitive.

As shown in Table I, Figs. 6 and 7, the superiority of the proposed algorithm is more evident for low-level noise than for high-level noise. The SNR and SSIM values of NNDCL1 are about 0.4dB and 0.01 higher than those of TVSCAD, respectively. However, with the increase of noise density, the superiority decreases gradually. This may be explained as follows. For low-level noise, both the regularization term and the data fidelity term play equally important roles. In this case, nonconvex regularizer has advantages over convex one. For high-level noise, such as 80% and 90% impulse noise, the data fidelity term plays a far more important role than the regularization term. Therefore, compared with TVSCAD, the superiority of NNDCL1 is no longer obvious. But, in general, by using the nonconvex log penalty to promote sparsity, NNDCL1 can obtain higher quality restored images than TVSCAD.

2) Compared With Nonconvex Models: In this part, the new approach is compared with some nonconvex methods [6], [8], [16]. In [6], ET function is used as data fitting term and TV is taken as regularization term. The model (4) is called ETTV and solved by proximal linearized minimization method. In [8], an L0TV nonconvex model (2) was presented and the proximal ADMM was used to solve the optimization problem. In [16], a nonconvex data fitting term plus nonconvex regularizer model (5) was introduced and the authors also presented an effective algorithm, called NNADM.

In this experiment, the same blur types are adopted as discussed in the last part. But, we test 30% and 80% noise levels of SP noise and 20% and 70% noise levels of RV noise. The tested images are Parrot (256 $\times$ 256), Lena (512 $\times$ 512), and Man (512 $\times$ 512), which are displayed in Fig. 8. The $\mu$ value of (6) is taken from {3.8, 15, 20, 75}, and $s$ is 0.035 for SP noise and 0.09 for RV noise. We choose penalty parameters $\beta_v, \beta_z, \beta_d \in \{2, 5, 10, 20, 50, 100, 650, 1000\}$ in the ADMM. The max outer iteration number is 5 or 10. For ETTV, L0TV, and NNADM, we use the suggested parameters values.

Numerical performances of various compared methods are recorded in Table II. The results show that NNDCL1 performs better than ETTV, L0TV, and NNADM in terms of SNR and SSIM in most cases. L0TV and NNADM seem to need more CPU time in seconds than ETTV and NNDCL1 except the case of 70% noise level. Moreover, the performances of ETTV and NNDCL1 are almost similar in terms of CPU times for all noise levels.

The visual comparisons are shown in Figs. 9–12. These figures give restored results of images degraded by average blur kernel with 30% SP noise, Gaussian blur kernel with 80% SP noise, Gaussian blur kernel with 20% RV noise and average blur kernel with 70% RV noise. We see that the four methods can remove the noise well, regardless of low or high-level noise.

The curves of partial SNR and SSIM versus CPU time are plotted in Figs. 13–16. Among the four algorithms, SNR and SSIM values of NNDCL1 are increasing as CPU time and always on the top at the last time. ETTV is in the middle of the four methods in most cases. From Figs. 13 and 14, we can find that the curves of NNADM are far below. The increasing rates of L0TV are slightly slower in Figs. 15 and 16. So, from these
TABLE II

<table>
<thead>
<tr>
<th>Images</th>
<th>Level</th>
<th>Blur</th>
<th>ETTV</th>
<th>L0TV</th>
<th>NNADM</th>
<th>NNDCL1</th>
<th>NNDCL1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>SNR</td>
<td>SNR</td>
<td>SNR</td>
<td>SNR</td>
<td>SNR</td>
</tr>
<tr>
<td>Images</td>
<td></td>
<td></td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
</tr>
<tr>
<td>Images</td>
<td></td>
<td></td>
<td>CPU</td>
<td>Time</td>
<td>CPU</td>
<td>Time</td>
<td>CPU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
</tr>
<tr>
<td>Images</td>
<td></td>
<td></td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
</tr>
<tr>
<td>Images</td>
<td></td>
<td></td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
</tr>
<tr>
<td>Images</td>
<td></td>
<td></td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
</tr>
<tr>
<td>Images</td>
<td></td>
<td></td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
<td>Value</td>
</tr>
</tbody>
</table>

Fig. 10. Gaussian blur image with 80% SP noise and images restored by ETTV, L0TV, NNADM, and NNDCL1.

Fig. 11. Gaussian blur image with 20% RV noise and images restored by ETTV, L0TV, NNADM, and NNDCL1.

Fig. 12. Average blur image with 70% RV noise and images restored by ETTV, L0TV, NNADM, and NNDCL1.

Fig. 13. SNR values versus CPU times for restoring Parrot, Lena, and Man images which are corrupted by Gaussian blur (first row) and average blur (second row) with 80% SP noise.

figures, we observe that NNDCL1 has a good convergence property and is stable.

B. Image Denoising

To further verify the performance of the proposed algorithm in reducing artifacts, image denoising (when \( K \) is the identity operator) experiment is tested. In this experiment, the proposed algorithm is compared with two very recent works \[26\], \[27\] for impulse noise removal. In \[26\], under the assumption that both signal and noise are sparse, the authors proposed a fast iterative method, called iterative double thresholding (IDT), to remove impulse noise, which has a low complexity. In \[27\], a nonconvex model called L0OGSTV was presented, which has an \( L_0 \) norm data fidelity term and overlapping group sparse TV (OGSTV) regularization function. This model can be considered as an extension of L0TV. Majorization-minimization method and ADMM were used to solve the \( L_0 \)-norm and the OGSTV optimization problem.

The test images are Shape image \( (256 \times 256) \) and MRI image \( (320 \times 320) \), as shown in Fig. 17. The SP and RV noises with different noise densities are tested. The noise levels of SP are 40% and 60%, and those of RV are 10% and 50%. For fairness, we manually chose the regularization and algorithm
Fig. 14. SSIM values versus CPU times for restoring Parrot, Lena, and Man images which are corrupted by Gaussian blur (first row) and average blur (second row) with 80% SP noise.

Fig. 15. SNR values versus CPU times for restoring Parrot, Lena, and Man images which are corrupted by Gaussian blur (first row) and average blur (second row) with 70% RV noise.

Fig. 16. SSIM values versus CPU times for restoring Parrot, Lena, and Man images which are corrupted by Gaussian blur (first row) and average blur (second row) with 70% RV noise.

Fig. 17. Tested images.

parameters which yields the best SNR and SSIM for IDT and NNDCL1, while for L0OGSTV, the parameter settings closely follow [27]. The maximum outer iteration of NNDCL1 takes 2 for low-level noise, and takes 3 or 5 for high-level noise.

The experiment results are reported in Table III. From the table, two observations can be made. First, the proposed NNDCL1 method almost always outperforms IDT and L0OGSTV methods for both SP and RV impulse noise removal. This is obvious especially in terms of SNR and SSIM. Second, IDT has the lowest runtime. The reason is that the main computational costs are two thresholding operators, and not involving the matrix–vector product.

We show the restored images in Figs. 18 and 19. From Fig. 18, we can see that all the test methods can remove the noise well when its density is low. Whereas, as shown in Fig. 19, IDT is not capable of completely removing the noise when the noise density is high or not sparse. Besides, although L0OGSTV preserves the details of the image, it tends to cause blurred edges or produce undesired artifacts, and its execution process usually takes a long time. By comparison, the proposed NNDCL1 method can achieve a good balance between artifact reduction and noise removal.

Besides, it is seen from the comparative experiment that the first iteration of Algorithm 1 produces a high-quality guess for the nonconvex optimization problem, and can be interpreted as a warm-start step. This means that the proposed DCA needs a less number of iterations to achieve the optimal solution, thus improving the overall efficiency of the proposed method.

C. Image Recovery

The third example is given to further show the superiority of the proposed method in image recovery with impulse noise and the effectiveness of the $L_1$ proximal technique. In this experiment, we apply our model to the image recovery from compressed measurements. The tested images are Shepp-Logan and Brain, and shown in Fig. 20. The size
TABLE III

<table>
<thead>
<tr>
<th>Images</th>
<th>SP (%)</th>
<th>SP Noise Rem.</th>
<th>RV (%)</th>
<th>RV Noise Rem.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>SNR</td>
<td>SSIM</td>
<td>Time</td>
<td>SNR</td>
</tr>
<tr>
<td>40%</td>
<td>13.031</td>
<td>14.3147</td>
<td>0.9641</td>
<td>99.2031</td>
</tr>
<tr>
<td>60%</td>
<td>20.5469</td>
<td>8.0463</td>
<td>0.8523</td>
<td>87.0311</td>
</tr>
<tr>
<td>MRI</td>
<td>10%</td>
<td>21.6503</td>
<td>14.8633</td>
<td>0.9144</td>
</tr>
<tr>
<td>50%</td>
<td>6.5626</td>
<td>7.7387</td>
<td>0.5919</td>
<td>105.9219</td>
</tr>
</tbody>
</table>

TABLE IV

<table>
<thead>
<tr>
<th>Noise</th>
<th>Images</th>
<th>L1LS-FISTA</th>
<th>$L_0.5$LS-ADMM</th>
<th>YALL1</th>
<th>$L_0.2$L-ADMM</th>
<th>$L_0.5$L-ADMM</th>
<th>$L_0.7$L-ADMM</th>
<th>$L_0.7$L-ADMM</th>
<th>NNDCL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian mixture</td>
<td>Logan 32.3301</td>
<td>32.5683</td>
<td>27.9568</td>
<td>39.3128</td>
<td>40.3454</td>
<td>39.2712</td>
<td>42.7397</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S\tau S$</td>
<td>Logan 12.5396</td>
<td>13.0372</td>
<td>29.5403</td>
<td>31.6272</td>
<td>32.7204</td>
<td>32.6292</td>
<td>33.5979</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Brain 12.1137</td>
<td>11.3154</td>
<td>24.8355</td>
<td>27.1307</td>
<td>27.5435</td>
<td>27.1410</td>
<td>27.2365</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 21. Recovery results of L1LS-FISTA, $L_0.5$LS-ADMM, YALL1, $L_0.2$L-ADMM, $L_0.5$L-ADMM, $L_0.7$L-ADMM, and NNDCL1 on Shepp-Logan and Brain images with Gaussian mixture noise.

of each image is $256 \times 256$. The measurement number is $m = \text{round}(0.4n)$, where $n = 65536$. A partial discrete cosine transformation matrix $A$ is employed as the sensing matrix $K$ and the Haar wavelets are used as the basis functions. Then, the model (6) becomes

$$\min_x \mu \Phi_f(Ax - b) + \Psi_s(x)$$

where $x$ is a wavelet coefficient and $b$ is the sampled wavelet data with impulse noise. We compare the new model with the $L_q$ regularized least absolute ($L_q$LA) formulation

$$\min_x \mu \|Ax - b\|_1 + \|x\|_q$$

and the $L_q$ regularized least squares ($L_q$LS) formulation

$$\min_x \mu \|Ax - b\|_2 + \|x\|_q$$

where $0 < q \leq 1$.

Two types of impulse noise [28] are tested. The first one is Gaussian mixture noise. The probability density function of two-term Gaussian model is given by

$$(1 - \epsilon)N(0, \sigma^2) + \epsilon N(0, \kappa \sigma^2)$$

where $\epsilon = 0.1$ and $\kappa = 1000$ in our experiments. The first term stands Gaussian thermal noise and the second term means the behavior of impulse noise. The second one is symmetric $\tau$-stable ($S\tau S$) noise. The characteristic function of $S\tau S$ distribution can be expressed as $e^{(\tau w) - \rho |w|^{\tau}}$, where the characteristic exponent $\tau$ is 1 and the scale parameter $\rho$ is $10^{-4}$. The smaller the value of $\tau$, the more impulse the noise is.

Our new method NNDCL1 is compared with some representative algorithms, which include L1LS-FISTA [29], $L_0$LS-ADMM [30], YALL1 [31], and $L_q$LA-ADMM [28]. L1LS-FISTA solves the $L_1$LS model. $L_0.5$LS-ADMM solves the $L_q$LS model based on ADMM with $q = 0.5$. YALL1 solves the robust $L_1$LA model. $L_q$LA-ADMM with different values of $q \in \{0.2, 0.5, 0.7\}$ solves the $L_q$LA model. The peak-signal noise ratio (PSNR) is used to evaluate the recovery performance.

Fig. 21 presents the recovery versions of the images corrupted by Gaussian mixture noise. We can see that all algorithms achieve good results in terms of visual quality. Fig. 22 gives the recovery images that have been damaged by $S\tau S$ noise. One can see that the $L_2$ data fidelity-based methods, that is, L1LS-FISTA and $L_0.5$LS-ADMM, are failed, while the $L_1$ data fidelity-based algorithms, that is, YALL1, $L_q$LA-ADMM and NNDCL1, work well. That is because the considered $S\tau S$ noise contains more impulse noise than Gaussian noise. The performances in terms of PSNR are shown in Table IV. The results show that NNDCL1 outperforms those representative algorithms on reconstructing Shepp-Logan. The improvements are higher as 2.39 dB (Gaussian mixture noise) and 0.87 dB ($S\tau S$ noise) over the compared algorithms. The reason may be that the $L_1$ norm proximal term of the proposed algorithm further promotes the sparsity in the iterations. This advantage
is decreased on recovering Brain image. The PSNR values attained by NNDCL1 are slightly lower than $L_0.5$LA-ADMM, but are higher than L1LS-FISTA, $L_0.5$LS-ADMM, YALL1, $L_0.5$LA-ADMM, $L_0.7$LA-ADMM, and NNDCL1 on Shepp-Logan and Brain images with SpS noise.

is decreased on recovering Brain image. The PSNR values attained by NNDCL1 are slightly lower than $L_0.5$LA-ADMM, but are higher than L1LS-FISTA, $L_0.5$LS-ADMM, YALL1, $L_0.5$LA-ADMM, and $L_0.7$LA-ADMM.

Table V reports the test results for Gaussian plus impulse noise removal. We can see from Table V that BdCNN achieves higher PSNR than NNDCL1 when the noise level is low. But, our method performs well in terms of PSNR and SSIM when the images are corrupted by heavy mixed noises. This is because our new method use the nonconvex data fitting term, which can detect impulse noise effectively. Hence, it can be seen that compared with the deep-learning-based method, our numerical results are quite competitive.

2) Gaussian Mixture Noise: In the second test, we compare the proposed method with EMCNN [33] on the removal of Gaussian mixture noise. EMCNN is a variational model for mixed noise removal, and is integrated with the CNN deep learning regularization. For EMCNN, image prior is learned by the CNN that is associated with a variational functional.

In this part, Pascal (500 test images, which are the same as those chosen in the previous test) and Berkeley Segmentation Dataset (BSD, 100 test images in total) datasets are used for the comparison. We consider the following Gaussian mixture noise form:

$$\epsilon N\left(0, \sigma_1^2\right) + (1 - \epsilon)N\left(0, \sigma_2^2\right).$$

Then, the test images are corrupted by Gaussian mixture noise with mixture ratio $\epsilon = 0.4, 0.5, 0.6$ and three noise levels, that is: 1) $\sigma_1 = 10$ and $\sigma_2 = 30$; 2) $\sigma_1 = 15$ and $\sigma_2 = 70$; and 3) $\sigma_1 = 25$ and $\sigma_2 = 60$. For EMCNN, the regularization trained by the authors of [33] is adopted, and the parameters are set as suggested in [33]. For the proposed method, the setting of the parameters is the same as that of the previous test.

Table VI reports the results for Gaussian mixed noise removal. It is seen from Table VI that the proposed method achieves the highest SSIM index except for the case of low noise level tested on BSD, which means that our method produces the best restored results in the most cases. We also observe that EMCNN performs better than NNDCL1 in terms of PSNR in the most cases. Fig. 23 shows the restored images of EMCNN and NNDCL1 for two test images (Bike image from Pascal and Penguin image from BSD) with different noise levels. We can see from Fig. 23 that EMCNN can retain more image details. Thus, the mean square error value of EMCNN is less than NNDCL1, which leads to that EMCNN has a higher PSNR value. However, a small amount of noise or artifacts still exist in the images restored by EMCNN. Although NNDCL1 generates oversmooth edges in the restored images, the noise is completely removed. By the
TABLE VI
AVERAGE PSNR AND SSIM VALUES: EMCNN AND NNDCL1 ON PASCAL AND BSD DATASETS

<table>
<thead>
<tr>
<th>DataSet</th>
<th>σ₁=10</th>
<th>σ₂=30</th>
<th>σ₁=15</th>
<th>σ₂=70</th>
<th>σ₁=25</th>
<th>σ₂=60</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pascal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NNDCL1</td>
<td>0.5834</td>
<td>0.7614</td>
<td>0.8071</td>
<td>0.6339</td>
<td>0.6784</td>
<td>0.7281</td>
</tr>
<tr>
<td>BSD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NNDCL1</td>
<td>0.7938</td>
<td>0.8039</td>
<td>0.8131</td>
<td>0.7416</td>
<td>0.7567</td>
<td>0.7699</td>
</tr>
</tbody>
</table>

Fig. 23. (a) Bike image, (b) noisy image, σ₁ = 25, σ₂ = 60, and ε = 0.4, (c) restored by EMCNN, PSNR = 26.1105, and SSIM = 0.6348, (d) restored by NNDCL1, PSNR = 23.0624, and SSIM = 0.7423, (e) Penguin image, (f) noisy image, σ₁ = 15, σ₂ = 70, and ε = 0.5, (g) restored by EMCNN, PSNR = 30.0007, and SSIM = 0.7337, and (h) restored by NNDCL1, PSNR = 29.3246, and SSIM = 0.7981.

Fig. 24. (a) Plane image, (b) noisy image with the level (20, 80, 30), (c) restored by BdCNN, PSNR = 15.0069, and SSIM = 0.5602, (d) restored by NNDCL1, PSNR = 20.6525, and SSIM = 0.8545, (e) Eagle image, (f) noisy image with the level (35, 80, 20), (g) restored by BdCNN, PSNR = 16.9129, and SSIM = 0.4900, and (h) restored by NNDCL1, PSNR = 19.2452, and SSIM = 0.6694.

In summary, as shown in Tables V and VI, it is obvious that our new method can effectively remove different levels of mixed noise. Though the tested deep-learning-based methods exhibit better performance in terms of PSNR, the proposed method has a higher SSIM value than EMCNN. Moreover, compared to the EMCNN, our new method shows an improvement in SSIM for high noise level. This shows the potential ability of our new method to improve the performance of high-level mixture noise removal.

VI. CONCLUSION
In this article, we have proposed a new model for image restoration with impulse noise. Although the resulting optimization problem is nonconvex, a novel and effective DCA has been designed for solving it. We have proved that the limit point of the sequence generated by the proposed algorithm is a stationary point of the nonconvex objective function. The convergence analysis expands the existing results in DC programming. Experimental results of image deblurring, denoising, and recovery demonstrated that the proposed approach is highly competitive compared with the existing popular methods. Note that it remains largely unexplored how to adaptively choose the regularization parameter μ. This is an open and challenging problem, and we leave the problem to further research.

REFERENCES


