International Journal of Engineering Research and Technology. ISSN 0974-3154 Vol.13, No.4 (2020), pp. 686-698
© International Research Publication House. https://dx.doi.org/10.37624/IJERT/13.4.2020.686-698

IMAGE DENOISING METHODS FOR NEW TVL1 MODELS WITH IMPULSE NOISE[†]

JAE HEON YUN*

Abstract:

In this paper, we first propose two TVL1 variational problems for restoring images degraded by impulse noise, and then we propose two fixed-point-like methods for solving the new proposed TVL1 problems. Numerical experiments for several test images corrupted by impulse noise and several damaged images overlying with texts are provided to demonstrate the efficiency and reliability of the proposed fixed-point-like methods. Numerical results show that two fixed-point-like methods for solving the new TVL1 problems perform significantly better in the image quality measured by PSNR than a fixed-point method for solving an existing TVL1 problem.

1. INTRODUCTION

In this paper, we consider image denoising problems degraded by impulse noise. Impulse noise is often generated by malfunctioning pixels in camera senses, faulty memory locations in hardware, or erroneous transmission. Two common types of impulse noise are salt-and-pepper noise and random-valued noise. Assume that an intensity range of an image is $[d_{min}, d_{max}]$. Salt-and-pepper noise corrupts a portion of pixels with only two values of d_{min} or d_{max} while keeping other pixels unaffected. For random-valued noise, a portion of pixels is corrupted in the same manner as salt-and-pepper noise except that the corrupted pixels can take any random value between d_{min} and d_{max}.

We first introduce the operator vec which transforms a matrix $C \in \mathbb{R}^{M\times N}$ into a long vector $c \in \mathbb{R}^{\ell}$ by stacking the columns of C from left to right, that is,

$$c = vec(C) = (c_{*1}^T, c_{*2}^T, \ldots, c_{*N}^T)^T \in \mathbb{R}^{\ell}$$

where c_{*i} is the ith column of C and $\ell = MN$. Let us assume that the true image U has an $N \times N$ square array. Then the original image U is represented by a long vector $u = vec(U)$ of size $m = N^2$. In this paper, we assume that a noisy image $f \in \mathbb{R}^m$ is represented by

$$f = u + \eta \qquad (1.1)$$

where $u \in \mathbb{R}^m$ is the original image and $\eta \in \mathbb{R}^m$ denotes the impulse noise. Then for salt-and-pepper noise, the noisy image $f = (f_i) \in \mathbb{R}^m$ is defined as

$$f_i = \begin{cases} d_{min} & \text{with probability } \frac{s}{2} \\ d_{max} & \text{with probability } \frac{s}{2} \\ u_i & \text{with probability } 1-s \end{cases},$$

where s is the noise level of the salt-and-pepper noise. Our objective of this paper is to restore u from the noisy image f as well as possible.

The classic TVL1 model for recovering a true image u from a noisy image f with impulse noise is given by the following variational problem with the l_1-norm data fidelity term and total variational regularization term

$$\min\left\{\|u-f\|_1 + \rho\|u\|_{\mathrm{TV}} : u \in \Gamma\right\}, \qquad (1.2)$$

where $\rho > 0$ is a regularization parameter, Γ is a closed convex subset of $\mathbb{R}^m$ and $\|u\|_{\mathrm{TV}}$ denotes the total variation (TV) of u. There are two possible definitions for $\|u\|_{\mathrm{TV}}$; one is the anisotropic TV, and the other is the isotropic TV. In this paper, we only consider the isotropic TV of $u \in \mathbb{R}^m$ which is defined by

$$\|u\|_{\mathrm{TV}} = \sum_{i=1}^{m} \|(\nabla u)_i\|_2 = \sum_{i=1}^{m} \left\| \begin{pmatrix} (\nabla_x u)_i \\ (\nabla_y u)_i \end{pmatrix} \right\|_2, \qquad (1.3)$$

where the discrete gradient operator $\nabla : \mathbb{R}^m \to \mathbb{R}^{2m}$ is defined as follows:

$$(\nabla u)_i = ((\nabla_x u)_i, (\nabla_y u)_i)^T \text{ for each } i = 1, 2, \ldots, m$$

with

$$(\nabla_x u)_i = \begin{cases} 0 & , \text{ if } i \bmod N = 1, \\ u_i - u_{i-1}, & \text{if } i \bmod N \neq 1, \end{cases}$$

$$\text{and } (\nabla_y u)_i = \begin{cases} 0 & , \text{ if } i \le N, \\ u_i - u_{i-N}, & \text{if } i > N. \end{cases}$$

Note that if ℓ_1-norm is used instead of ℓ_2-norm in Equation (1.3), then the resulting TV norm is anisotropic. The convex subset Γ of $\mathbb{R}^m$ associated with a given vector $f \in \mathbb{R}^m$ and a proper subset Λ of $\{1, 2, \ldots, m\}$ is given by

$$\Gamma = \{\, y \in \mathbb{R}^m \mid M_\Lambda\, y = M_\Lambda\, f \,\}, \qquad (1.4)$$

1991 *Mathematics Subject Classification.* 94A08, 54E05, 49Q20, 35M85.

Key words and phrases. TVL1 problems, fixed-point-like method, image denoising, impulse noise, image inpainting.

[†] This work was supported by the National Research Foundation of Korea(NRF) funded by the Korea government(MSIT) (No. 2019R1F1A1060718).

* Corresponding author. Department of Mathematics, College of Natural Sciences, Chungbuk National University, Cheongju, Korea 28644. E-mail: gmjae@chungbuk.ac.kr.

where M_Λ is an $m \times m$ diagonal matrix with diagonal entries 1 for indices in Λ and 0 otherwise.

In last decades, the problem of solving the classical TVL1 model (1.2) has been studied by many researchers (see [2, 3, 4, 5, 6, 8, 10, 11, 12] and the references therein). It was shown in [5, 8, 10, 11] that the TVL1 model works successfully in recovering blurred images corrupted by impulse noise. Notice that the TVL1 model has many difficulties in finding its solutions both mathematically and numerically since both the l_1-norm data fidelity and regularization terms are not differentiable.

Recently, Lu et al. [7] proposed a fixed-point method for solving the following TVL1 variational problem

$$\min\left\{\|u-f\|_1+\frac{\lambda}{2}\|u\|_2^2+\rho\|u\|_{\mathrm{TV}} : u \in \Gamma\right\}, \qquad (1.5)$$

where λ and ρ are positive numbers. They showed that the fixed point method performs remarkably better in the image quality measured by PSNR and preserves more features than FTVd (Fast total variation deconvolution) proposed in [11] at the expense of much increase in computational time. This approach motivates us to propose the following two TVL1 variational problems

$$\min\left\{\|u-f\|_1+\frac{\lambda}{2}\|Du\|_2^2+\rho\|u\|_{\mathrm{TV}} : u \in \Gamma\right\}, \qquad (1.6)$$

$$\min\left\{\|u-f\|_1+\lambda\|Du\|_2+\rho\|u\|_{\mathrm{TV}} : u \in \Gamma\right\}, \qquad (1.7)$$

where $D = -\Delta$ and Δ denotes a discrete Laplacian operator. Under the reflexive boundary condition, the discrete Laplacian operator is represented by a singular matrix in $\mathbb{R}^{m\times m}$ (see Section 6).

Notice that the TVL1 problem (1.5) has a unique solution since its objective function is strictly convex, while the TVL1 problems (1.6) and (1.7) may not have a unique solution since its objective functions are just convex, not strictly convex. By using the indicated function of the convex set Γ which is defined by

$$\iota_\Gamma(u)=\begin{cases}0 & \text{if } u\in\Gamma\\ \infty & \text{if } u\notin\Gamma\end{cases},$$

the constrained minimization problems (1.5), (1.6) and (1.7) can be transformed into the following equivalent unconstrained minimization problems

$$\min\left\{\|u-f\|_1+\frac{\lambda}{2}\|u\|_2^2+\rho\|u\|_{\mathrm{TV}}+\iota_\Gamma(u) : u \in \mathbb{R}^m\right\}, \qquad (1.8)$$

$$\min\left\{\|u-f\|_1+\frac{\lambda}{2}\|Du\|_2^2+\rho\|u\|_{\mathrm{TV}}+\iota_\Gamma(u) : u \in \mathbb{R}^m\right\}, \qquad (1.9)$$

$$\min\left\{\|u-f\|_1+\lambda\|Du\|_2+\rho\|u\|_{\mathrm{TV}}+\iota_\Gamma(u) : u \in \mathbb{R}^m\right\}. \qquad (1.10)$$

This paper is organized as follows. In Section 2, we provide some definitions and useful properties which are fundamental tools for developing numerical algorithms for solving the image denoising problems (1.6) and (1.7). In Section 3, we briefly review the fixed-point method for the TVL1 problem (1.5) proposed by Lu et al. [7]. In Section 4, we propose a fixed-point-like method, using proximal operators, for solving the new proposed TVL1 problem (1.6). In Section 5, we propose a fixed-point-like method, using proximal operators, for solving the new proposed TVL1 problem (1.7). In Section 6, we provide numerical experiments for several test images corrupted by salt-and-pepper impulse noise and several damaged images overlying with texts in order to demonstrate the efficiency and reliability of two fixed-point-like methods for solving the TVL1 problems (1.6) and (1.7). Lastly we provide some concluding remarks.

2. PRELIMINARIES

Some definitions and useful results which we refer to later in this paper are provided below. We first provide the proximal operator introduced by Moreau [9].

Definition 2.1. Let $\psi : \mathbb{R}^m \to \mathbb{R}\cup\{+\infty\}$ be a proper, convex and lower semi-continuous (l.s.c) function. The *proximal operator* of ψ at $v \in \mathbb{R}^m$ is defined by

$$\mathrm{prox}_\psi(v)=\arg\min\left\{\frac{1}{2}\parallel u-v\parallel_2^2+\psi(u) : u\in\mathbb{R}^m\right\}. \qquad (2.1)$$

Definition 2.2. Let $\psi : \mathbb{R}^m \to \mathbb{R}\cup\{+\infty\}$ be a proper, convex and l.s.c function. The *subdifferential* of ψ at $v \in \mathbb{R}^m$ is defined by

$$\partial\psi(v)=\{y\in\mathbb{R}^m : \psi(z)\ge\psi(v)+<y,z-v>,\ \forall z\in\mathbb{R}^m\}. \qquad (2.2)$$

Elements in $\partial\psi(v)$ are called *subgradeients*.

It is well-known that subdifferential of a convex function ψ is a set-valued mapping from $\mathbb{R}^m$ into a nonempty convex compact set in $\mathbb{R}^m$ [1]. We now present three examples for which we can explicitly calculate the proximal operators.

Example 2.3. Let $||\cdot||_1$ and $||\cdot||_2$ be the l_1-norm and l_2-norm defined on $\mathbb{R}^m$, respectively. For any $\lambda > 0$ and $v \in \mathbb{R}^m$

$$\mathrm{prox}_{\frac{1}{\lambda}\|\cdot\|_1}(v)=\max\left\{|v|-\frac{1}{\lambda},0\right\}.*\,\mathrm{sign}(v),$$

$$\mathrm{prox}_{\frac{1}{\lambda}\|\cdot\|_2}(v)=\max\left\{\|v\|_2-\frac{1}{\lambda},\ 0\right\}\frac{v}{\|v\|_2},$$

where $|v|$ denotes elementwise absolute value of the vector v and $.*$ denotes the elementwise multiplication.

Example 2.4. Let Γ be the convex subset of $\mathbb{R}^m$ which is defined by (1.4). For any $\lambda > 0$ and $v \in \mathbb{R}^m$

$$\mathrm{prox}_{\frac{1}{\lambda}\iota_\Gamma}(v)=M_\Lambda f+(I-M_\Lambda)(v),$$

where ι_Γ denotes the indicated function of the convex set Γ and I denotes an identity matrix of order m.

Notice that the isotropic TV of $u \in \mathbb{R}^m$ defined by (1.3) can be expressed as

$$\|u\|_{\mathrm{TV}} = (\varphi \circ B)(u), \tag{2.3}$$

where $\varphi : \mathbb{R}^{2m} \to \mathbb{R}$ is a convex function defined by

$$\varphi(v) = \sum_{i=1}^{m} \left\| \begin{pmatrix} v_i \\ v_{m+i} \end{pmatrix} \right\|_2 \quad \text{for each } v = (v_i) \in \mathbb{R}^{2m}$$

and B is a $d \times m$ matrix which represents a discrete gradient operator $\triangledown$ with $m = N^2$ and $d = 2m$ (see Section 6). The next example gives the proximal operator of the convex function $\psi = \frac{1}{\lambda}\varphi$ on $\mathbb{R}^{2m}$ which is called the generalized shrinkage formula, where $\lambda > 0$.

Example 2.5. If $\psi = \frac{1}{\lambda}\varphi$ and $v = (v_i) \in \mathbb{R}^{2m}$, then

$$\mathrm{prox}_{\frac{1}{\lambda}\varphi}(v) = \prod_{i=1}^{m} \left(\mathrm{prox}_{\frac{1}{\lambda}\|\cdot\|_2} \begin{pmatrix} v_i \\ v_{m+i} \end{pmatrix} \right),$$

where $\prod$ denotes Cartesian product of vector spaces.

The following theorem outlines a relationship between the proximal operator and the subdifferential of a convex function.

Theorem 2.6 ([7, 9])**.** *If ψ is a proper, convex and l.s.c. function on $\mathbb{R}^m$ and $v \in \mathbb{R}^m$, then*

$$y \in \partial\psi(v) \;\Leftrightarrow\; v = \mathrm{prox}_{\psi}(v+y) \;\Leftrightarrow\; y = (I - \mathrm{prox}_{\psi})(v+y), \tag{2.4}$$

where I denotes an identity operator on $\mathbb{R}^m$.

3. REVIEW OF FIXED-POINT ALGORITHM FOR (1.5)

In this section, we briefly review the fixed-point method proposed in [7] for solving the TVL1 problem (1.5). The problem (1.5), which is equivalent to (1.8), can be expressed as

$$\min\left\{ \|u - f\|_1 + \frac{\lambda}{2}\|u\|_2^2 + \rho(\varphi \circ B)(u) + \iota_\Gamma(u) : u \in \mathbb{R}^m \right\}, \tag{3.1}$$

where $\lambda > 0$, $\rho > 0$, and φ and B are defined the same as in (2.3). Using Fermat rule in convex analysis for model (3.1) and relation $\partial(\varphi \circ B) = B^T \circ (\partial\varphi) \circ B$, we have

$$0 \in \left(\partial\frac{1}{\lambda}\|\cdot\|_1\right)(u - f) + \rho B^T \left(\partial\frac{1}{\lambda}\varphi\right)(Bu) + u + \frac{1}{\lambda}\partial\iota_\Gamma(u). \tag{3.2}$$

From relation (3.2), for any α, β, $\gamma > 0$ we can choose a vector $a \in \partial(\frac{1}{\alpha\lambda}\|\cdot\|_1)(u - f)$, a vector $b \in \partial(\frac{1}{\beta\lambda}\varphi)(Bu)$ and a vector $c \in \partial(\frac{1}{\gamma}\iota_\Gamma)(u)$ satisfying

$$\alpha a + \rho\beta B^T b + u + \frac{\gamma}{\lambda}c = 0. \tag{3.3}$$

By Theorem 2.6, the inclusions $a \in \partial(\frac{1}{\alpha\lambda}\|\cdot\|_1)(u - f)$, $b \in \partial(\frac{1}{\beta\lambda}\varphi)(Bu)$ and $c \in \partial(\frac{1}{\gamma}\iota_\Gamma)(u)$ lead to

$$a = \left(I - \mathrm{prox}_{\frac{1}{\alpha\lambda}\|\cdot\|_1}\right)(u - f + a), \tag{3.4}$$

$$b = \left(I - \mathrm{prox}_{\frac{1}{\beta\lambda}\varphi}\right)(Bu + b), \tag{3.5}$$

$$u = \mathrm{prox}_{\frac{1}{\gamma}\iota_\Gamma}(u + c). \tag{3.6}$$

Note that $\iota_\Gamma(\cdot) = \delta\,\iota_\Gamma(\cdot)$ for any $\delta > 0$. Setting $\gamma = \lambda$ in Equation (3.3) yields

$$c = -u - \alpha a - \rho\beta B^T b. \tag{3.7}$$

Substituting (3.7) into (3.6) and then applying Example 2.4 to Equation (3.6), one obtains

$$u = M_\Lambda f + (I - M_\Lambda)(-\alpha a - \rho\beta B^T b). \tag{3.8}$$

From Equations (3.4), (3.5) and (3.8), Lu et al. [7] proposed the following fixed point method, called Algorithm 1, for solving the TVL1 problem (1.5).

Algorithm 1 Fixed-point method for the TVL1 problem (1.5)

```
Given degraded image $f$, choose positive parameters $\alpha, \beta, \lambda, \rho$
Initialization : $u^0 = 0$, $a^0 = 0$ and $b^0 = 0$
for $k = 0$ to $maxit$ do
    $a^{k+1} = \left(I - \mathrm{prox}_{\frac{1}{\alpha\lambda}\|\cdot\|_1}\right)\left(u^k - f + a^k\right)$
    $b^{k+1} = \left(I - \mathrm{prox}_{\frac{1}{\beta\lambda}\varphi}\right)\left(Bu^k + b^k\right)$
    $u^{k+1} = M_\Lambda f + (I - M_\Lambda)\left(-\alpha a^{k+1} - \rho\beta B^T b^{k+1}\right)$
    if $\frac{\|u^{k+1} - u^k\|_2}{\|u^{k+1}\|_2} < tol$ then
        Stop
    end if
end for
```

For all algorithms considered in this paper, $maxit$ denotes the maximum number of iterations and tol denotes the tolerance value of the stopping criterion.

4. FIXED-POINT-LIKE METHOD FOR THE TVL1 PROBLEM (1.6)

In this section, we propose a fixed-point-like method, using the proximal operators, for solving the new proposed TVL1 variational problem (1.6). The problem (1.6), which is equivalent to (1.9), can be expressed as

$$\min\left\{ \|u - f\|_1 + \frac{\lambda}{2}\|Du\|_2^2 + \rho(\varphi \circ B)(u) + \iota_\Gamma(u) : u \in \mathbb{R}^m \right\}, \tag{4.1}$$

where $\lambda > 0$, $\rho > 0$, and φ and B are defined the same as in (2.3). Using Theorem 2.6, we can obtain the following property for a solution to the TVL1 problem (4.1).

Theorem 4.1. *If φ is a real-valued convex function on $\mathbb{R}^d$, B is an $d \times m$ matrix, and u is a solution to the TVL1 problem (4.1), then for any $\alpha, \beta > 0$ there exist vectors $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^d$ such that*

$$a = \left(I - \mathrm{prox}_{\frac{1}{\alpha}\|\cdot\|_1}\right)(u - f + a), \tag{4.2}$$

$$b = \left(I - \mathrm{prox}_{\frac{\rho}{\beta}\varphi}\right)(Bu + b), \tag{4.3}$$

$$u = M_\Lambda f + (I - M_\Lambda)\big(u - a - \frac{1}{\alpha}(\lambda D^T Du + \beta B^T b)\big). \tag{4.4}$$

Conversely, if there exist positive numbers α, β, $a \in \mathbb{R}^m$, $b \in \mathbb{R}^d$ and $u \in \mathbb{R}^m$ satisfying Equations (4.2) to (4.4), then u is a solution to the TVL1 problem (4.1).

Proof. We assume that $u \in \mathbb{R}^m$ is a solution to the TVL1 problem (4.1). By the Fermat rule in convex analysis for problem (4.1) and using the relations $\partial(\varphi \circ B) = B^T \circ (\partial\varphi) \circ B$, we have

$$0 \in (\partial\|\cdot\|_1)(u - f) + \lambda D^T Du + \rho B^T(\partial\varphi)(Bu) + \partial\iota_\Gamma(u). \tag{4.5}$$

From relation (4.5), for any $\alpha, \beta, \gamma > 0$ we can choose a vector $a \in \partial(\frac{1}{\alpha}\|\cdot\|_1)(u - f)$, a vector $b \in \partial(\frac{\rho}{\beta}\varphi)(Bu)$ and a vector $c \in \partial(\frac{1}{\gamma}\iota_\Gamma)(u)$ satisfying

$$\alpha\, a + \lambda D^T Du + \beta B^T b + \gamma\, c = 0. \tag{4.6}$$

By Theorem 2.6, the inclusions $a \in \partial(\frac{1}{\alpha}\|\cdot\|_1)(u-f)$ and $b \in \partial(\frac{\rho}{\beta}\varphi)(Bu)$ lead to Equations (4.2) and (4.3), respectively. The inclusion $c \in \partial(\frac{1}{\gamma}\iota_\Gamma)(u)$ means

$$u = \mathrm{prox}_{\frac{1}{\gamma}\iota_\Gamma}(u + c). \tag{4.7}$$

Setting $\gamma = \alpha$ in Equation (4.6) yields

$$c = -a - \frac{1}{\alpha}(\lambda D^T Du + \beta B^T b). \tag{4.8}$$

Substituting (4.8) into (4.7) and then applying Example 2.4 to Equation (4.7), Equation (4.4) is obtained.

Conversely, suppose that there exist α, β, $a \in \mathbb{R}^m$, $b \in \mathbb{R}^d$ and $u \in \mathbb{R}^m$ satisfying Equations (4.2) to (4.4). Again, by Theorem 2.6, Equations (4.2) and (4.3) ensure that $a \in \partial(\frac{1}{\alpha}\|\cdot\|_1)(u - f)$ and $b \in \partial(\frac{\rho}{\beta}\varphi)(Bu)$, respectively. In addition, Equation (4.4) means $u = \mathrm{prox}_{\frac{1}{\alpha}\iota_\Gamma}\big(u - a - \frac{1}{\alpha}(\lambda D^T Du + \beta B^T b)\big)$. Then Theorem 2.6 implies

$$-a - \frac{1}{\alpha}(\lambda D^T Du + \beta B^T b) \in \partial(\frac{1}{\alpha}\iota_\Gamma)(u). \tag{4.9}$$

From relation (4.9), it can be easily seen that relation (4.5) holds. Hence $u \in \mathbb{R}^m$ is a solution to the TVL1 problem (4.1). □

From Equations (4.2) to (4.4) of Theorem 4.1, we can obtain a fixed-point-like method, called Algorithm 2, using the proximal operators for the TVL1 problem (1.6).

Algorithm 2 Fixed-point-like method for the TVL1 problem (1.6)

```
 1: Given degraded image $f$, choose positive parameters $\alpha, \beta, \lambda, \rho$
 2: Initialization : $u^0 = 0$, $a^0 = 0$ and $b^0 = 0$
 3: for $k = 0$ to $maxit$ do
 4:     $a^{k+1} = \big(I - \mathrm{prox}_{\frac{1}{\alpha}\|\cdot\|_1}\big)\big(u^k - f + a^k\big)$
 5:     $b^{k+1} = \big(I - \mathrm{prox}_{\frac{\rho}{\beta}\varphi}\big)\big(Bu^k + b^k\big)$
 6:     $u^{k+1} = M_\Lambda f + (I - M_\Lambda)\big(u^k - a^{k+1} - \frac{1}{\alpha}(\lambda D^T Du^k + \beta B^T b^{k+1})\big)$
 7:     if $\frac{\|u^{k+1} - u^k\|_2}{\|u^{k+1}\|_2} < tol$ then
 8:         Stop
 9:     end if
10: end for
```

The following theorem provides a convergence analysis for Algorithm 2.

Theorem 4.2. *Let $\{a^n\}$, $\{b^n\}$ and $\{u^n\}$ be sequences generated by Algorithm 2. If we can find two consecutive vectors u^k and u^{k+1} such that $u^{k+1} = u^k$ for some positive values of α, β, λ and ρ, then u^{k+1} is a solution to the TVL1 problem (1.6).*

Proof. Substituting Equations (4.2) and (4.3) into Equation (4.4), one obtains

$$\begin{aligned} u = M_\Lambda f + (I - M_\Lambda)\Big(u - \big(I - \mathrm{prox}_{\frac{1}{\alpha}\|\cdot\|_1}\big)(u - f + a) \\ - \frac{1}{\alpha}\Big(\lambda D^T Du + \beta B^T\big(I - \mathrm{prox}_{\frac{\rho}{\beta}\varphi}\big)(Bu + b)\Big)\Big). \end{aligned} \tag{4.10}$$

From Theorem 4.1, it can be easily seen that if u, a and b satisfy (4.10) for some positive values of α, β, λ and ρ, then u is a solution to the TVL1 problem (1.6). In Algorithm 2, if $u^{k+1} = u^k$ for some positive values of α, β, λ and ρ, then we have

$$a^{k+1} = \big(I - \mathrm{prox}_{\frac{1}{\alpha}\|\cdot\|_1}\big)\big(u^{k+1} - f + a^k\big), \tag{4.11}$$

$$b^{k+1} = \big(I - \mathrm{prox}_{\frac{\rho}{\beta}\varphi}\big)\big(Bu^{k+1} + b^k\big). \tag{4.12}$$

Substituting Equations (4.11) and (4.12) into line 6 of Algorithm 2, one obtains

$$\begin{aligned} u^{k+1} = M_\Lambda f + (I - M_\Lambda)\Big(u^{k+1} - \big(I - \mathrm{prox}_{\frac{1}{\alpha}\|\cdot\|_1}\big)(u^{k+1} - f + a^k) \\ - \frac{1}{\alpha}\Big(\lambda D^T Du^{k+1} + \beta B^T\big(I - \mathrm{prox}_{\frac{\rho}{\beta}\varphi}\big)(Bu^{k+1} + b^k)\Big)\Big). \end{aligned} \tag{4.13}$$

From Equation (4.13), it can be seen that u^{k+1}, a^k and b^k satisfy (4.10) for some positive values of α, β, λ and ρ. Hence u^{k+1} is a solution to the TVL1 problem (1.6). □

Theorem 4.2 gives an idea of how to stop Algorithm 2. In practical applications, we do not have to find u^{k+1} which is equal to u^k. Instead, we need to find u^{k+1} which is reasonably close to u^k. Hence we have used the following stopping criterion in Algorithm 2

$$\frac{\|u^{k+1}-u^k\|_2}{\|u^{k+1}\|_2} < tol,$$

where tol is a suitably chosen small tolerance value.

5. FIXED-POINT-LIKE METHOD FOR THE TVL1 PROBLEM (1.7)

In this section, we propose a fixed-point-like method, using the proximal operators, for solving the new proposed TVL1 variational problem (1.7). The problem (1.7), which is equivalent to (1.10), can be expressed as

$$\min\left\{\|u-f\|_1+\lambda\|Du\|_2+\rho(\varphi\circ B)(u)+\iota_\Gamma(u) : u\in\mathbb{R}^m\right\}, \tag{5.1}$$

where $\lambda > 0$, $\rho > 0$, and φ and B are defined the same as in (2.3). Using Theorem 2.6, we can obtain the following property for a solution to the TVL1 problem (5.1).

Theorem 5.1. *If φ is a real-valued convex function on $\mathbb{R}^d$, B is an $d\times m$ matrix, and u is a solution to the TVL1 problem (5.1), then for any $\alpha,\beta,\gamma > 0$ there exist vectors $a,c\in\mathbb{R}^m$ and $b\in\mathbb{R}^d$ such that*

$$a=\left(I-\operatorname{prox}_{\frac{1}{\alpha}\|\cdot\|_1}\right)(u-f+a), \tag{5.2}$$

$$b=\left(I-\operatorname{prox}_{\frac{\rho}{\beta}\varphi}\right)(Bu+b), \tag{5.3}$$

$$c=\left(I-\operatorname{prox}_{\frac{1}{\gamma}\|\cdot\|_2}\right)(Du+c), \tag{5.4}$$

$$u=M_\Lambda f+(I-M_\Lambda)\big(u-a-\frac{1}{\alpha}(\lambda\gamma D^Tc+\beta B^Tb)\big). \tag{5.5}$$

Conversely, if there exist positive numbers α,β,γ and vectors $a,c\in\mathbb{R}^m$, $b\in\mathbb{R}^d$, and $u\in\mathbb{R}^m$ satisfying Equations (5.2) to (5.5), then u is a solution to the TVL1 problem (5.1).

Proof. We assume that $u\in\mathbb{R}^m$ is a solution to the TVL1 problem (5.1). By the Fermat rule in convex analysis for problem (5.1), we can obtain

$$0\in(\partial\|\cdot\|_1)(u-f)+B^T\left(\partial\rho\varphi\right)(Bu)+\lambda D^T(\partial\|\cdot\|_2)(Du)+\partial\iota_\Gamma(u). \tag{5.6}$$

From relation (5.6), for any $\alpha,\beta,\gamma > 0$ we can choose a vector $a\in\partial(\frac{1}{\alpha}\|\cdot\|_1)(u-f)$, $b\in\partial(\frac{\rho}{\beta}\varphi)(Bu)$, $c\in\partial(\frac{1}{\gamma}\|\cdot\|_2)(Du)$ and $d\in\partial(\frac{1}{\delta}\iota_\Gamma)(u)$ satisfying

$$\alpha\, a+\beta B^Tb+\gamma\lambda D^Tc+\delta\, d=0. \tag{5.7}$$

Hence Equation (5.5) holds. By Theorem 2.6, the inclusions $a\in\partial(\frac{1}{\alpha}\|\cdot\|_1)(Ku-f)$, $b\in\partial(\frac{\rho}{\beta}\varphi)(Bu)$ and $c\in\partial(\frac{1}{\gamma}\|\cdot\|_2)(Du)$ lead to Equations (5.2), (5.3) and (5.4), respectively. The inclusion $d\in\partial(\frac{1}{\delta}\iota_\Gamma)(u)$ means

$$u=\operatorname{prox}_{\frac{1}{\delta}\iota_\Gamma}(u+c). \tag{5.8}$$

Setting $\delta=\alpha$ in Equation (5.7) yields

$$d=-a-\frac{1}{\alpha}(\lambda\gamma D^Tc+\beta B^Tb). \tag{5.9}$$

Substituting (5.9) into (5.8) and then applying Example 2.4 to Equation (5.8), Equation (5.5) is obtained.

Conversely, suppose that there exist $\alpha,\beta,\gamma > 0$, $a,c,u\in\mathbb{R}^m$, and $b\in\mathbb{R}^d$ satisfying Equations (5.2) to (5.5). By Theorem 2.6, Equations (5.2) to (5.4) ensure that $a\in\partial(\frac{1}{\alpha}\|\cdot\|_1)(u-f)$, $b\in\partial(\frac{\rho}{\beta}\varphi)(Bu)$ and $c\in\partial(\frac{1}{\gamma}\|\cdot\|_2)(Du)$, respectively. In addition, Equation (5.5) means $u=\operatorname{prox}_{\frac{1}{\alpha}\iota_\Gamma}\big(u-a-\frac{1}{\alpha}(\lambda\gamma D^Tc+\beta B^Tb)\big)$. Then Theorem 2.6 implies

$$-a-\frac{1}{\alpha}(\lambda\gamma D^Tc+\beta B^Tb)\in\partial(\frac{1}{\alpha}\iota_\Gamma)(u). \tag{5.10}$$

From relation (5.10), it can be easily seen that relation (5.6) holds. Hence $u\in\mathbb{R}^m$ is a solution to the TVL1 problem (5.1). □

From Equations (5.2) to (5.5) of Theorem 5.1, we can obtain a fixed-point-like method, called Algorithm 3, using the proximal operators for the TVL1 problem (1.7).

Algorithm 3 Fixed-point-like method for the TVL1 problem (1.7)

```
Given degraded image $f$, choose positive parameters $\alpha,\beta,\gamma,\lambda,\rho$
Initialization : $u^0=0$, $a^0=0$, $b^0=0$ and $c^0=0$
for $k=0$ to $maxit$ do
    $a^{k+1}=(I-\operatorname{prox}_{\frac{1}{\alpha}\|\cdot\|_1})(u^k-f+a^k)$
    $b^{k+1}=(I-\operatorname{prox}_{\frac{\rho}{\beta}\varphi})(Bu^k+b^k)$
    $c^{k+1}=(I-\operatorname{prox}_{\frac{1}{\gamma}\|\cdot\|_2})(Du^k+c^k)$
    $u^{k+1}=M_\Lambda f+(I-M_\Lambda)(u^k-a^{k+1}-\frac{1}{\alpha}(\lambda\gamma D^Tc^{k+1}+\beta B^Tb^{k+1}))$
    if $\frac{\|u^{k+1}-u^k\|_2}{\|u^{k+1}\|_2} < tol$ then
        Stop
    end if
end for
```

The following theorem provides a convergence analysis for Algorithm 3.

Theorem 5.2. *Let $\{a^n\}$, $\{b^n\}$, $\{c^n\}$ and $\{u^n\}$ be sequences generated by Algorithm 3. If we can find two consecutive vectors u^k and u^{k+1} such that $u^{k+1}=u^k$ for some positive values of $\alpha,\beta,\gamma,\lambda$ and ρ, then u^{k+1} is a solution to the TVL1 problem (1.7).*

Proof. Substituting Equations (5.2) to (5.4) into Equation (5.5), one obtains

$$\begin{aligned} u=M_\Lambda f+(I-M_\Lambda)\Big(u-&\big(I-\operatorname{prox}_{\frac{1}{\alpha}\|\cdot\|_1}\big)(u-f+a)\\ &-\frac{1}{\alpha}\Big(\lambda\gamma D^T\big(I-\operatorname{prox}_{\frac{1}{\gamma}\|\cdot\|_2}\big)(Du+c)\\ &+\beta B^T\big(I-\operatorname{prox}_{\frac{\rho}{\beta}\varphi}\big)(Bu+b)\Big)\Big). \end{aligned} \tag{5.11}$$

Theorem 5.1 implies that if u, a, b and c satisfy (5.11) for some positive values of $\alpha,\beta,\gamma,\lambda$ and ρ, then u is a solution to the TVL1 problem (1.7). In Algorithm 3, if $u^{k+1} = u^k$ for some positive values of $\alpha, \beta, \gamma, \lambda$ and ρ, then we have

$$a^{k+1} = \big(I - \mathrm{prox}_{\frac{1}{\alpha}\|\cdot\|_1}\big)\big(u^{k+1} - f + a^k\big), \tag{5.12}$$

$$b^{k+1} = \big(I - \mathrm{prox}_{\frac{\rho}{\beta}\varphi}\big)\big(Bu^{k+1} + b^k\big), \tag{5.13}$$

$$c^{k+1} = \big(I - \mathrm{prox}_{\frac{1}{\gamma}\|\cdot\|_2}\big)\big(Du^{k+1} + c^k\big). \tag{5.14}$$

Substituting Equations (5.12) to (5.14) into line 7 of Algorithm 3, one obtains

$$\begin{aligned} u^{k+1} = {} & M_\Lambda f + (I - M_\Lambda)\Big(u^{k+1} - (I - \mathrm{prox}_{\frac{1}{\alpha}\|\cdot\|_1})(u^{k+1} - f + a^k) \\ & - \frac{1}{\alpha}\Big(\lambda\gamma D^T (I - \mathrm{prox}_{\frac{1}{\gamma}\|\cdot\|_2})\Big(Du^{k+1} + c^k\Big) \\ & + \beta B^T (I - \mathrm{prox}_{\frac{\rho}{\beta}\varphi})(Bu^{k+1} + b^k)\Big)\Big). \end{aligned} \tag{5.15}$$

Equation (5.15) implies that u^{k+1}, a^k, b^k and c^k satisfy (5.11) for some positive values of $\alpha, \beta, \gamma, \lambda$ and ρ. Hence u^{k+1} is a solution to the TVL1 problem (1.7). □

Theorem 5.2 also gives an idea of how to stop Algorithm 3. Thus Algorithm 3 has the same stopping criterion as Algorithm 2.

6. NUMERICAL EXPERIMENTS

In this section, we provide numerical performance results for Algorithms 1 to 3. In Section 2, it was shown that the isotropic TV of $u \in \mathbb{R}^m$ can be represented by $||u||_{\mathrm{TV}} = (\varphi \circ B)(u)$, where φ and B are defined the same as in (2.3). Let $D_x = D_y$ be the first order backward finite difference matrix of order N defined by

$$D_x = D_y = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ -1 & 1 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -1 & 1 & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{pmatrix}.$$

Then B can be expressed as a $d \times m$ matrix given by

$$B = \begin{pmatrix} I_N \otimes D_x \\ D_y \otimes I_N \end{pmatrix},$$

where $\otimes$ denotes the Kronecker product, I_N denotes the identity matrix of order N, $m = N^2$ and $d = 2m$. Under the reflexive boundary condition, the discrete negative Laplacian operator $D = -\Delta$ can be represented by a singular $m \times m$ matrix $(I_N \otimes D_{xx} + D_{yy} \otimes I_N)$, where $D_{xx} = D_{yy}$ is the second order finite difference matrix of order N defined by

$$D_{xx} = D_{yy} = \begin{pmatrix} 1 & -1 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 1 \end{pmatrix}.$$

In order to illustrate the efficiency and reliability of two fixed-point-like methods, called Algorithms 2 and 3, for solving the new proposed TVL1 problems (1.6) and (1.7), we provide numerical results for four test images such as Cameraman, Lena, House and Boat (see Figure 1). The pixel size of four test images is 256×256. All numerical tests have been performed using Matlab R2019a on a personal computer with 3.2GHz CPU and 8GB RAM. $maxit$ is set to 500 for all algorithms, and tol is set to 1×10^{-5} (for Algorithm 1), 1×10^{-4} (for Algorithm 2) or 2×10^{-3} (for Algorithm 3).

To evaluate the quality of the restored images, we use the peak signal-to-noise ratio (PSNR) between the restored image and original image which is defined by

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{N^2 \cdot \max_{i,j}|u_{ij}|^2}{\|U - \tilde{U}\|_F^2}\right)$$

where $\|\cdot\|_F$ refers to the Frobenius norm, U and $\tilde{U}$ are the original and restored images with size $N \times N$, respectively. Also u_{ij} stands for the value of original image U at the pixel point (i, j) and N^2 is the total number of pixels. It is generally true that the larger PSNR value stands for the better quality of restored image.

For all numerical experiments, we have used the test images with an intensity range of $[0, 1]$. For all test problems, we choose the degraded test images which are corrupted by salt-and-pepper impulse noise with noise levels 60% or 80%. In Tables 1 to 4, P_0 represents the PSNR values for the noisy images f, Alg denotes the algorithm to be used, Cam denotes the Cameraman image, *PSNR* represents the PSNR value for the restored image, *Iter* denotes the number of iterations, and *CPU* denotes the elapsed CPU time in seconds. All parameters α, β, γ, λ and ρ are chosen as the best one by numerical tries. The convex set Γ is generated from an index set Λ which contains indices corresponding to the intact part of the original image $u = vec(U)$.

Tables 1 and 2 contain numerical results of Algorithms 1 to 3 for noisy images with 60% and 80% salt-and-pepper noises, respectively. Tables 3 and 4 contain numerical results of Algorithms 1 to 3 for two different types of overlying images. Figure 1 shows the true images for Cameraman, Lena, House and Boat. Figures 2 and 3 show the images restored by Algorithms 1 to 3 for noisy images with 60% and 80% salt-and-pepper noise, respectively. Figures 4 and 5 show the images inpainted by Algorithms 1 to 3 for two different types of overlying images, respectively.

TABLE 1. Numerical results for image denoising problems with 60% salt-and-pepper noise

Image	P_0	Alg	α	β	γ	λ	ρ	tol	$PSNR$	$Iter$	CPU
Cam	7.21	1	0.0001	0.0001		0.001	2500	1×10^{-5}	26.08	55	0.81
		2	150	3.5		4.9	2.0	1×10^{-4}	26.70	167	3.86
		3	600	250	1.25	0.4	30	2×10^{-3}	26.78	208	5.19
Lena	7.20	1	0.0001	0.0001		0.001	2500	1×10^{-5}	27.91	54	0.81
		2	150	3.5		4.9	2.0	1×10^{-4}	28.51	155	3.58
		3	600	250	1.25	0.4	30	2×10^{-3}	28.56	148	3.65
House	7.13	1	0.0001	0.0001		0.001	2500	1×10^{-5}	31.62	55	0.82
		2	150	3.5		4.9	2.0	1×10^{-4}	33.18	180	4.15
		3	600	250	1.25	0.4	30	2×10^{-3}	33.42	237	5.88
Boat	6.85	1	0.0001	0.0001		0.001	2500	1×10^{-5}	26.87	55	0.82
		2	150	3.5		4.9	2.0	1×10^{-4}	27.49	159	3.66
		3	600	250	1.25	0.4	30	2×10^{-3}	27.70	175	4.36

TABLE 2. Numerical results for image denoising problems with 80% salt-and-pepper noise

Image	P_0	Alg	α	β	γ	λ	ρ	tol	$PSNR$	$Iter$	CPU
Cam	5.95	1	0.0001	0.0001		0.001	2500	1×10^{-5}	23.30	120	1.82
		2	150	3.5		4.7	2.0	1×10^{-4}	23.71	315	7.20
		3	600	250	1.25	0.4	30	2×10^{-3}	23.78	260	6.38
Lena	5.93	1	0.0001	0.0001		0.001	2500	1×10^{-5}	25.10	116	1.74
		2	150	3.5		4.7	2.0	1×10^{-4}	25.57	291	6.65
		3	600	250	1.25	0.4	30	2×10^{-3}	25.84	186	4.54
House	5.84	1	0.0001	0.0001		0.001	2500	1×10^{-5}	27.96	123	1.87
		2	150	3.5		4.7	2.0	1×10^{-4}	28.98	363	8.27
		3	600	250	1.25	0.4	30	2×10^{-3}	29.62	329	8.17
Boat	5.59	1	0.0001	0.0001		0.001	2500	1×10^{-5}	23.80	119	1.80
		2	150	3.5		4.7	2.0	1×10^{-4}	23.99	291	6.67
		3	600	250	1.25	0.4	30	2×10^{-3}	24.28	206	5.15

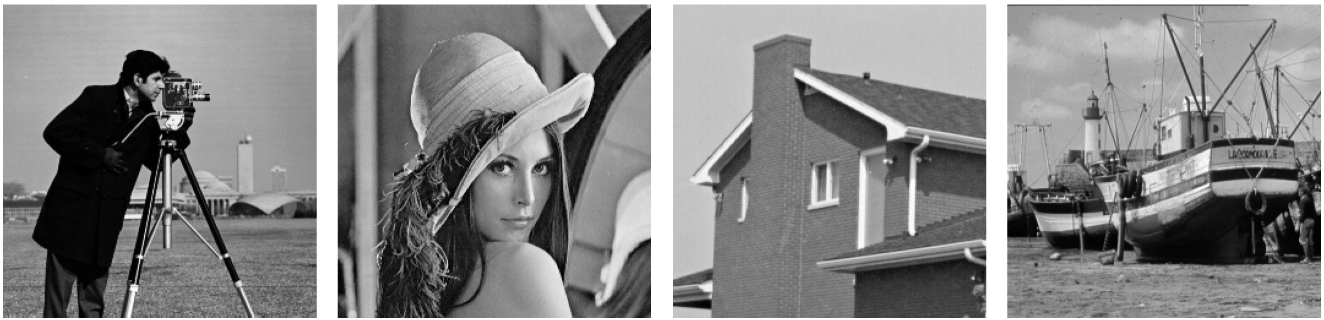
TABLE 3. Numerical results for image inpainting problems with overlying images beginning with words "The term"

Image	P_0	Alg	α	β	γ	λ	ρ	tol	$PSNR$	$Iter$	CPU
Cam	18.50	1	0.0001	0.0001		0.001	2500	1×10^{-5}	36.77	52	0.72
		2	150	50		2.0	16	1×10^{-4}	37.14	200	4.24
		3	150	50	0.12	2.0	13.5	2×10^{-3}	37.84	237	5.47
Lena	16.86	1	0.0001	0.0001		0.001	2500	1×10^{-5}	37.49	52	0.68
		2	150	50		2.0	16	1×10^{-4}	38.11	163	3.44
		3	150	50	0.12	2.0	13.5	2×10^{-3}	39.14	173	3.98
House	16.89	1	0.0001	0.0001		0.001	2500	1×10^{-5}	40.32	52	0.67
		2	150	50		2.0	16	1×10^{-4}	40.34	244	5.17
		3	150	50	0.12	2.0	13.5	2×10^{-3}	40.62	286	6.59
Boat	15.74	1	0.0001	0.0001		0.001	2500	1×10^{-5}	35.62	51	0.67
		2	150	50		2.0	16	1×10^{-4}	35.58	189	3.99
		3	150	50	0.12	2.0	13.5	2×10^{-3}	36.44	185	4.27

TABLE 4. Numerical results for image inpainting problems with overlying images beginning with words "Cross-Correlation"

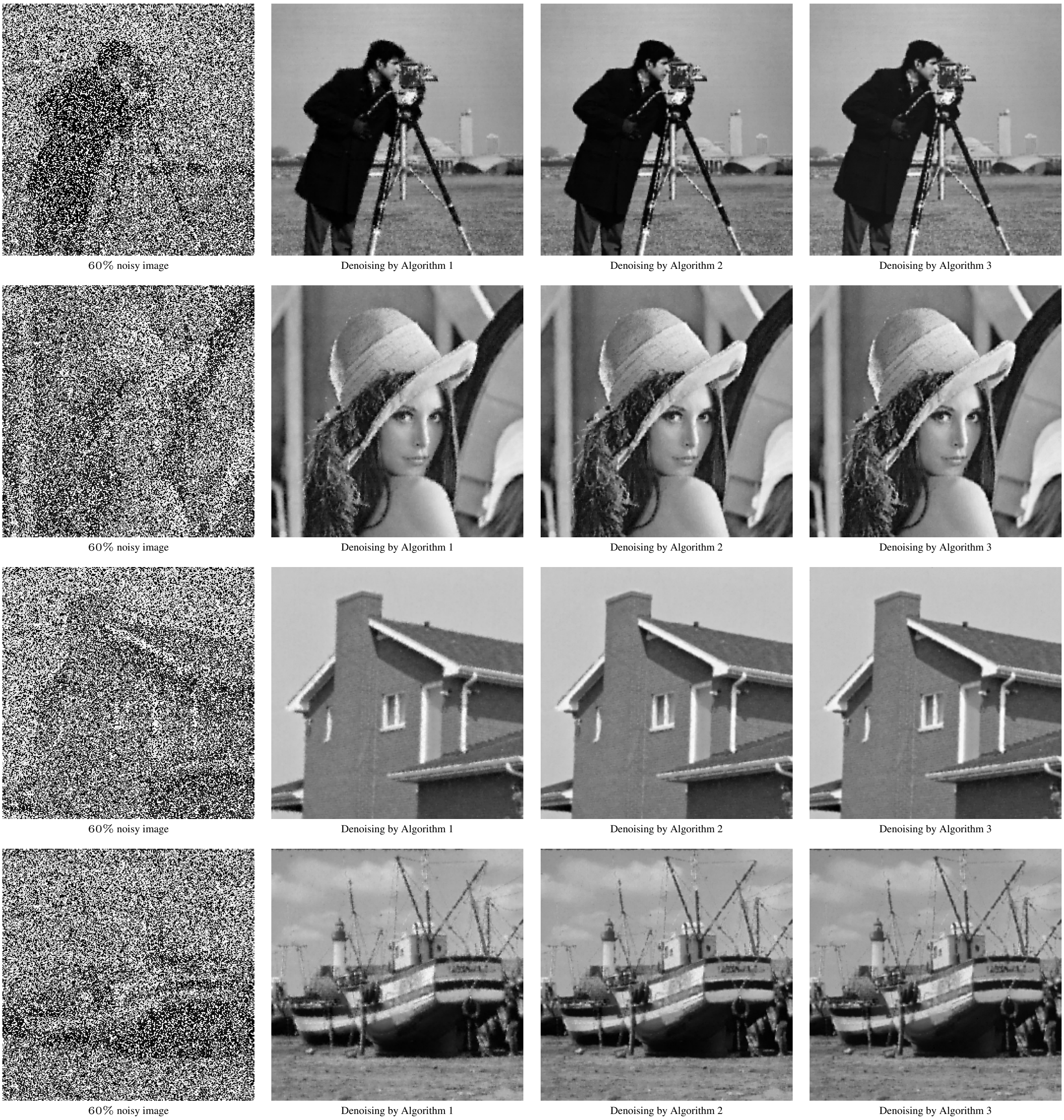
Image	P_0	Alg	α	β	γ	λ	ρ	tol	$PSNR$	$Iter$	CPU
Cam	16.81	1	0.0001	0.0001		0.001	2500	1×10^{-5}	34.35	30	0.38
		2	150	50		2.0	16	1×10^{-4}	34.47	153	3.28
		3	150	50	0.12	2.0	13.5	2×10^{-3}	35.13	188	4.33
Lena	17.26	1	0.0001	0.0001		0.001	2500	1×10^{-5}	38.95	30	0.38
		2	150	50		2.0	16	1×10^{-4}	39.43	121	2.57
		3	150	50	0.12	2.0	13.5	2×10^{-3}	39.57	133	3.08
House	18.46	1	0.0001	0.0001		0.001	2500	1×10^{-5}	41.26	30	0.37
		2	150	50		2.0	16	1×10^{-4}	42.77	168	3.58
		3	150	50	0.12	2.0	13.5	2×10^{-3}	42.88	227	5.24
Boat	17.42	1	0.0001	0.0001		0.001	2500	1×10^{-5}	37.57	30	0.37
		2	150	50		2.0	16	1×10^{-4}	37.26	146	3.11
		3	150	50	0.12	2.0	13.5	2×10^{-3}	38.59	163	3.76

As can be seen in Tables 1 to 4, Algorithm 3 for the TVL1 problem (1.7) restores the true image best, and Algorithm 1 for the TVL1 problem (1.5) restores the true image worst in almost all cases. That is, Algorithm 3 yields the highest PSNR values, and Algorithm 1 yields the lowest PSNR values in almost all cases. Algorithm 3 for the TVL1 problem (1.7) restores the true image significantly better than Algorithm 2 for the TVL1 problem (1.6), but Algorithm 3 takes more CPU time than Algorithm 2 except for 80% salt-and-pepper noise. Also observe that Algorithm 1 restores the true image worse than Algorithms 2 and 3, but it takes much less CPU time than Algorithms 2 and 3. Based on numerical results, Algorithm 3 for solving the new TVL1 problem (1.7) is preferred over Algorithms 1 and 2.



Cameraman image Lena image House image Boat image

FIGURE 1. True images for Cameraman, Lena, House and Boat



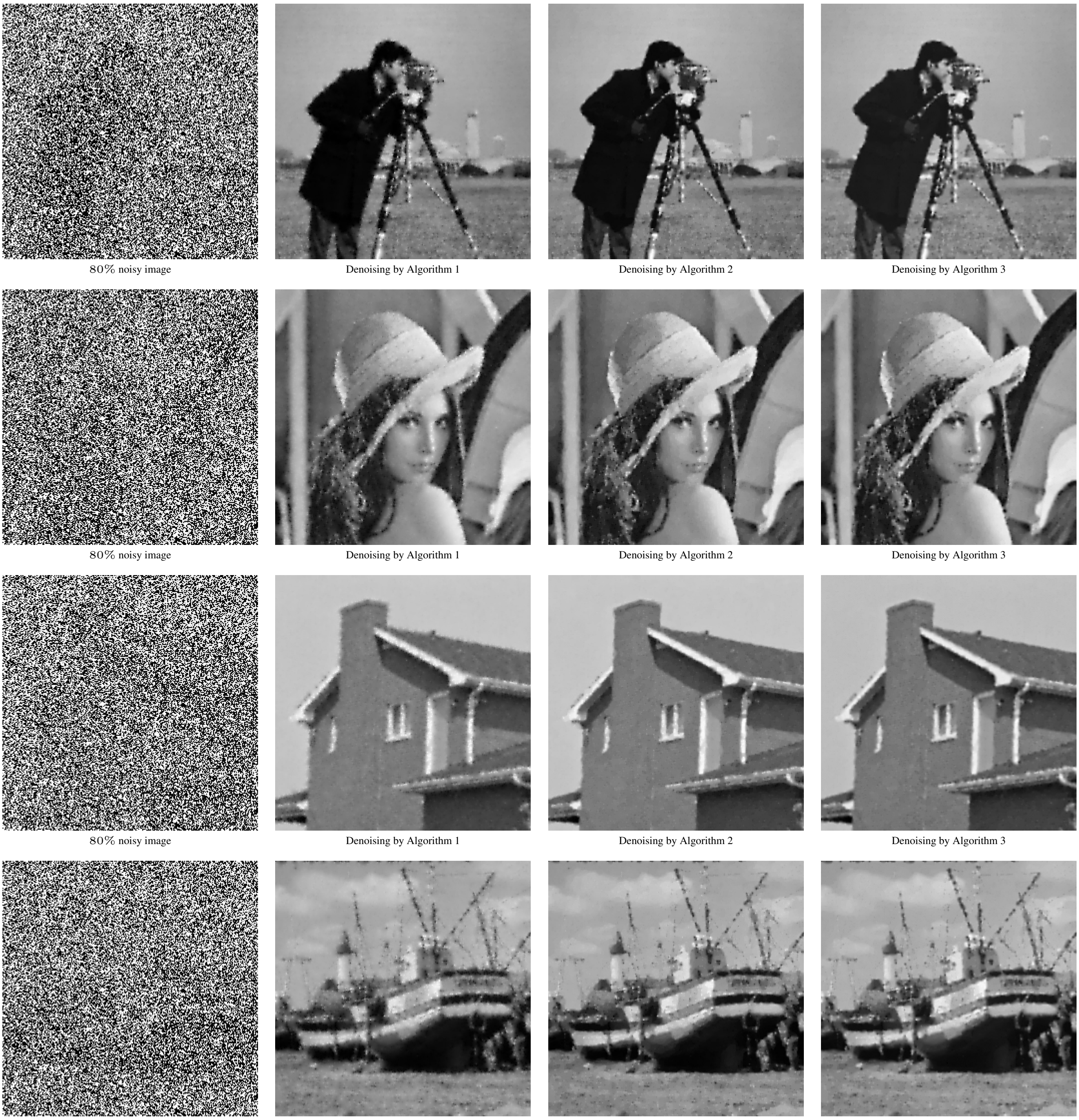
60% noisy image Denoising by Algorithm 1 Denoising by Algorithm 2 Denoising by Algorithm 3

60% noisy image Denoising by Algorithm 1 Denoising by Algorithm 2 Denoising by Algorithm 3

60% noisy image Denoising by Algorithm 1 Denoising by Algorithm 2 Denoising by Algorithm 3

60% noisy image Denoising by Algorithm 1 Denoising by Algorithm 2 Denoising by Algorithm 3

FIGURE 2. Image denoising for noisy images with 60% salt-and-pepper noise



80% noisy image

Denoising by Algorithm 1

Denoising by Algorithm 2

Denoising by Algorithm 3

80% noisy image

Denoising by Algorithm 1

Denoising by Algorithm 2

Denoising by Algorithm 3

80% noisy image

Denoising by Algorithm 1

Denoising by Algorithm 2

Denoising by Algorithm 3

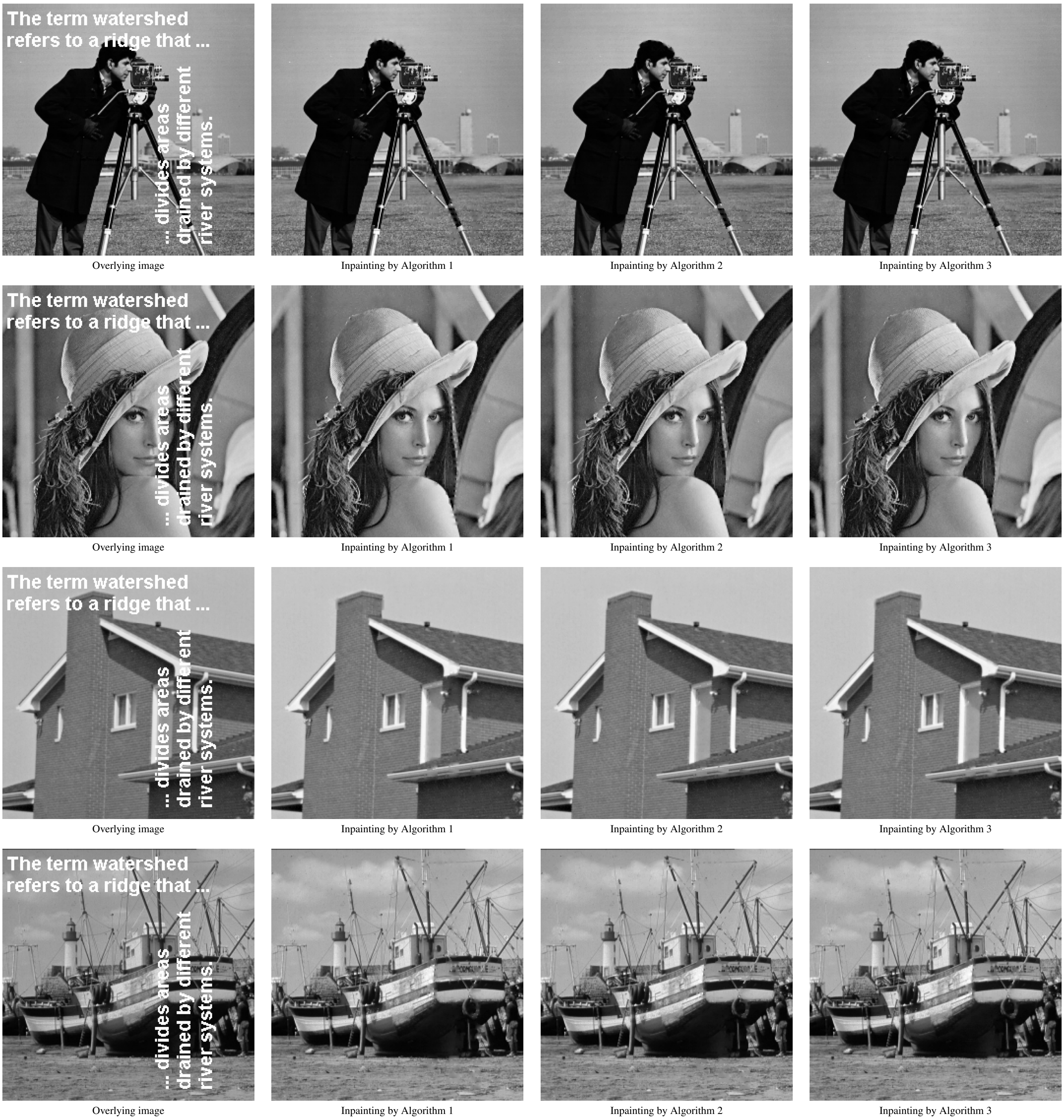
80% noisy image

Denoising by Algorithm 1

Denoising by Algorithm 2

Denoising by Algorithm 3

FIGURE 3. Image denoising for noisy images with 80% salt-and-pepper noise



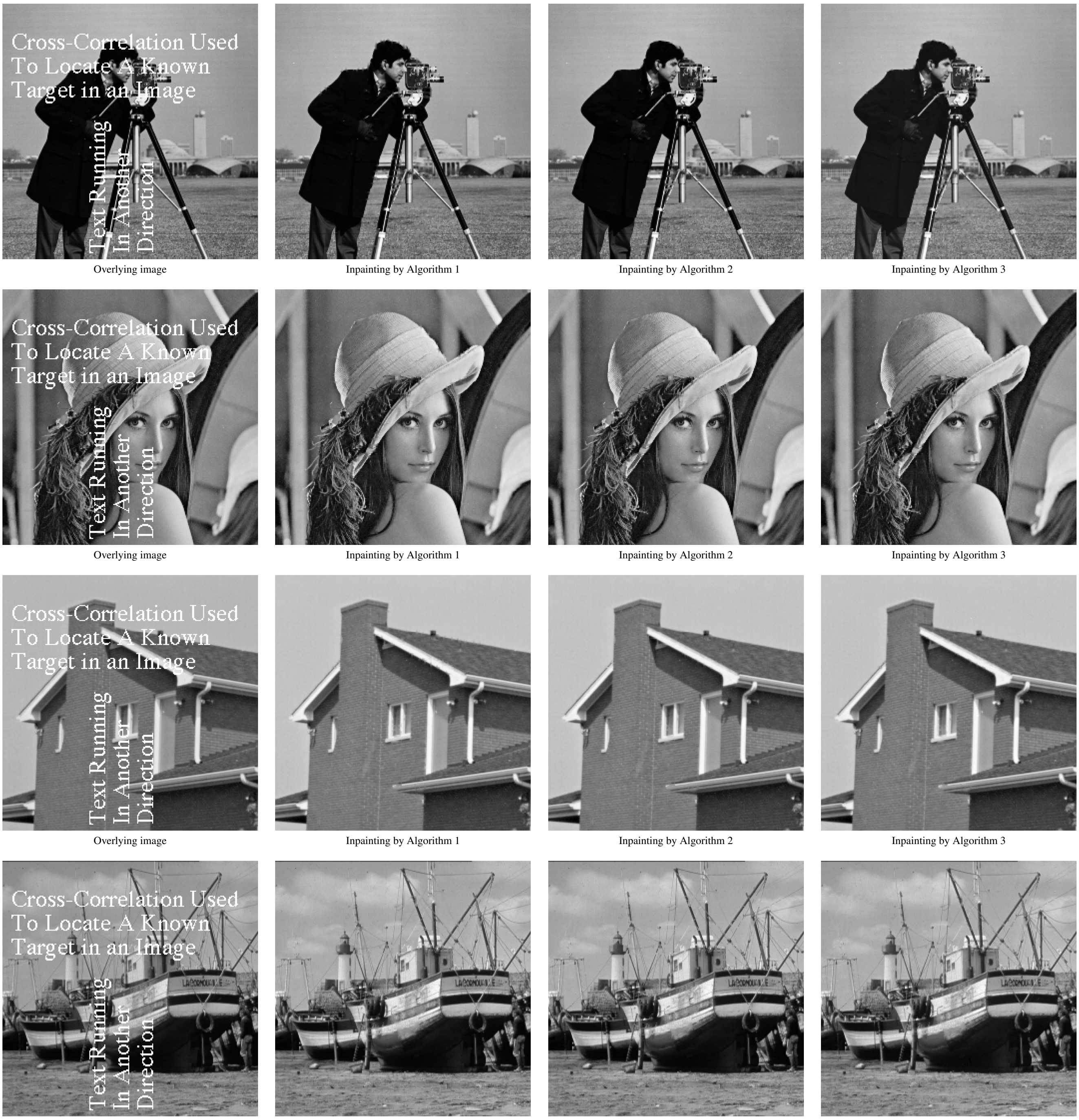
Overlying image Inpainting by Algorithm 1 Inpainting by Algorithm 2 Inpainting by Algorithm 3

Overlying image Inpainting by Algorithm 1 Inpainting by Algorithm 2 Inpainting by Algorithm 3

Overlying image Inpainting by Algorithm 1 Inpainting by Algorithm 2 Inpainting by Algorithm 3

Overlying image Inpainting by Algorithm 1 Inpainting by Algorithm 2 Inpainting by Algorithm 3

FIGURE 4. Image inpainting for damaged images overlying with texts



Overlying image Inpainting by Algorithm 1 Inpainting by Algorithm 2 Inpainting by Algorithm 3

Overlying image Inpainting by Algorithm 1 Inpainting by Algorithm 2 Inpainting by Algorithm 3

Overlying image Inpainting by Algorithm 1 Inpainting by Algorithm 2 Inpainting by Algorithm 3

Overlying image Inpainting by Algorithm 1 Inpainting by Algorithm 2 Inpainting by Algorithm 3

FIGURE 5. Image inpainting for damaged images overlying with texts

7. CONCLUSION

In this paper, we proposed two new TVL1 variational problems (1.6) and (1.7) for restoring images degraded by impulse noise, and then we proposed two fixed-point-like methods, called Algorithms 2 and 3, for solving the new TVL1 problems (1.6) and (1.7). Numerical experiments showed that Algorithms 2 and 3 for solving the new proposed TVL1 problems (1.6) and (1.7) perform better in the image quality measured by PSNR than Algorithm 1 for solving the existing TVL1 problem (1.5). In addition, Algorithm 3 restores the true image significantly better than Algorithm 2, but Algorithm 3 takes more CPU time than Algorithm 2 for most cases. Hence, it can be concluded that Algorithm 3 for solving the new TVL1 problem (1.7) is preferred over Algorithms 1 and 2.

REFERENCES

[1] A. Beck, First-order methods in optimization, SIAM, Philadelphia PA, 2017.

[2] T.F. Chan and S. Esedoglu, *Aspects of total variation regularized L1 function approximation*, SIAM J. Appl. Math., **65** (2005), 1817–1837.

[3] Y. Dong, M. Hintermüller and M. Neri, *An efficient primal–dual method for L1-TV image restoration*, SIAM J. Imag. Sci., **2** (2009), 1168–1189.

[4] V. Duval, J.-F. Aujol and Y. Gousseau, *The TVL1 model: A geometric point of view*, SIAM J. Multiscale Model. Simul., **8** (2009), 154–189.

[5] Y.D. Han and J.H. Yun, *Performance of Fixed-point-like Methods for a TVL1 Problem with Impulse Noise*, International Journal of Engineering Research and Technology, **12** (2019), 2431–2439.

[6] Q. Li, C.A. Micchelli, L. Shen and Y. Xu, *A proximity algorithm accelerated by Gauss–Seidel iterations for L1/TV denoising models*, Inverse Problems, **28** (2012), Article ID 095003, p. 20.

[7] J. Lu, K. Qiao, L. Shen and Y. Zou, *Fixed-point algorithms for a TVL1 image restoration model*, Inte. J. Comp. Math., **95** (2018), 1829–1844.

[8] C.A. Micchelli, L. Shen, Y. Xu and X. Zeng, *Proximity algorithms for the L1/TV image denoising model*, Adv. Comput. Math. **38** (2013), 401–426.

[9] J.J. Moreau, *Proximité et dualité dans un espace Hilbertien*, Bull. Soc. Math. Fr. Mém., **93** (1965), 273–299.

[10] M. Nikolova, *A variational approach to remove outliers and impulse noise*, J. Math. Imaging and Vision, **20** (2004), 99–120.

[11] J.F. Yang, Y. Zhang and W.T. Yin, *An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise*, SIAM J. Sci. Comput., **31** (2008), 2842–2865.

[12] W. Yin, D. Goldfarb and S. Osher, *The total variation regularized L1 model for multiscale decomposition*, SIAM J. Multiscale Model. Simul., **6** (2007), 190–211.